UNIVERSITÉ DU QUÉBEC À MONTRÉAL

L'EXPLOITATION DE L'AUTO-SIMILARITÉ POUR LA PRÉDICTION DU TRAFIC INTERNET

MÉMOIRE
PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INFORMATIQUE

PAR
MAHER CHTIOUI

MARS 2006

REMERCIEMENTS

Les travaux présentés dans ce rapport ont été effectués au Laboratoire Téléinformatique du Département d'Informatique de l'université de Québec à Montréal, dirigé par Mme. Halima ElBiaze, que je tiens à remercier cordialement pour son accueil et pour avoir partagé avec moi ses connaissances et m'avoir prodigué ses conseils et qui a pu m'épauler et répondre à toutes mes questions pendant toute la durée de mon stage.

J'adresse mes plus vifs remerciements à Monsieur Omar Cherkaoui, Directeur et responsable du laboratoire téléinformatique à L'UQÀM pour m'avoir permis d'intégrer l'équipe de recherche.

Je tiens à remercier très chaleureusement Monsieur Hafedh Mili, professeur au Département d'Informatique à l'UQÀM pour son encouragement et son soutien permanent tout au long de mes études.

De plus, je voudrais remercier le secrétariat pour sa disponibilité ainsi que mes collègues, ceux du laboratoire téléinformatique ainsi que tous les autres, pour m'avoir accueilli et accepté dans leur groupe, pour avoir permis que ce stage me soit autant profitable sur le plan professionnel que personnel.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

RÉSUMÉ

Les recherches qui se font pour modéliser l'évolution de trafic Internet sur une grande échelle de temps et pour développer des modèles de prédictions à court et à long terme constituent les domaines les plus intéressés par les chercheurs en technologies de l'information. En effet, pour compléter et affiner la caractérisation du trafic introduite par les premières études et recherches, il est fort utile de modéliser de façon nouvelle et efficace le trafic Internet. Jusqu'à présent, les ingénieurs et chercheurs en réseau utilisaient des modèles Mathématiques et Statistiques pour modéliser les processus de trafic. Pour ce, il faudrait proposer de nouveaux modèles de trafic réalistes. On peut notamment citer le trafic réseau le plus réaliste en intégrant les notions d'auto-similarité et de dépendance à long terme LRD. En utilisant des outils mathématiques, on a commencé par chercher le type du processus à partir des historiques de l'information et savoir le type de trafic pour des différentes échelles de temps, ceci exige une grande collection de mesures sur une longue durée de temps. Ce que nous a permit de montrer que le trafic dans un réseau Internet est exposé à des fortes périodicités et des variabilités aux multiples échelles de temps.

Une autre contribution importante, décrite en détail dans ce mémoire, consiste à utiliser les résultats des simulations de trafic Internet plus réalistes afin d'appliquer des algorithmes de prédiction sur ces séries simulées. Quatre algorithmes ont été testés dans ce cadre intégrant des modèles mathématiques afin d'offrir des preuves et des validations du bon fonctionnement des solutions proposées, pour en sortir à la fin, le meilleur algorithme qui permet de donner un taux d'erreurs minimum ainsi qu'une grande simplicité dans son application.

CHAPITRE I

INTRODUCTION

La modélisation de trafic est très importante pour l'étude, l'analyse et la conception des réseaux que se soit téléinformatique, routier ou n'importe quel autre trafic. On s'intéresserait au trafic dans les réseaux téléinformatique surtout avec l'introduction de nouvelles applications qui ont donnée naissance à d'autres hypothèses des caractéristiques de trafic.

Un bon modèle du trafic de réseaux permettra une meilleure conception de protocoles ainsi qu'une bonne topologie et architecture de réseaux.

Le changement des caractéristiques du trafic dans les réseaux est dû essentiellement au grand nombre de machines interconnectées dans le réseau, ainsi qu'à l'utilisation de plusieurs applications hétérogènes comme l'Internet, le Telnet, la voix, l'image, etc...

Ces applications ont causé une augmentation importante du volume d'échanges de données sur le réseau. Ces nouveaux types de trafic ont pu changer toutes les caractéristiques du trafic des données dans le réseau d'ordinateurs. Et comme résultat de ces observations et de ces nouvelles tendances, les recherches ont été augmentées dans le domaine de la caractérisation du trafic ainsi que leurs implications dans la conception des ordinateurs, et la configuration des réseaux téléinformatiques.

Parmi les nouvelles caractéristiques du trafic dans les réseaux est la dépendance à mémoire longue (LRD) qu'on retrouve dans le Web, les réseaux locaux

Ethernet ainsi que les réseaux métropolitains. L'autre caractéristique importante est celle d'auto–similarité (Self-Similarity) que nous allons détailler ces notions dans les prochains paragraphes.

Le chercheur Partridge [28] a pu donner une explication générale sur les nouvelles caractéristiques de trafic réseau : « *On n'a pas encore compris le comportement du trafic des données en communication. Après un quart de siècle de communication, les chercheurs sont dans l'incapacité de fournir un modèle adéquat pour le trafic. Aujourd'hui on doit prendre des décisions concernant la façon de configurer les réseaux et de construire des composantes basées sur des modèles non adéquats.* »

Les travaux de ces dernières années sur la modélisation du trafic des paquets dans l'Internet ont montré qu'un trait important concernant la nature du trafic Internet est son aspect auto - similaire (ou possédant des dépendances à long terme, ou encore de caractère fractal, sans trop entrer dans les subtilités mathématiques qui différencient ces trois notions voisines). L'auto - similarité du trafic de données a été mise en évidence dans [5]: la structure des variations d'amplitude du signal analysé (par exemple le nombre d'octets ou de paquets transférés par unité de temps ou la série des durées inter-paquets) se reproduit de manière similaire quelle que soit la finesse temporelle avec laquelle il est représenté. Le comportement d'un trafic auto - similaire est à l'opposé de celui d'un trafic poissonnien, pour lequel les variations d'amplitude sont filtrées au fur et à mesure que l'on augmente la taille de la fenêtre d'intégration.

Une monographie entière [32] récemment publiée est consacrée à la modélisation du phénomène d'autosimilarité du trafic dans les réseaux de données et à son impact sur l'évaluation des performances. Y figurent aussi des chapitres sur la description et la simulation des caractéristiques auto-similaires du trafic.

Cette autosimilarité, voire cette multi-fractalité, et son extrême variabilité à toutes les échelles de temps sont une caractéristique du principe de la commutation de paquets qui induit des transmissions en rafales [6]. Ce comportement se caractérise notamment par une décroissance lente, par exemple sous forme de loi de puissance,

sous exponentielle ou à queue lourde [7], de la fonction d'auto-corrélation du nombre de paquets transférés par unité de temps (typiquement 100 ms) : les processus auto-similaires sont des cas particuliers des processus à dépendance à long terme (LRD).

Même pour les applications 'stream', la caractéristique de LRD qui se retrouve dans les transferts de séquences vidéo à débit variable (VBR selon la terminologie ATM) [8], sont probablement due à la variabilité des paramètres de transmission liés au codage des trames (MPEG, par exemple), et à la dynamique des images, etc.

Concernant le trafic de type élastique, l'identification du processus des paquets tel qu'il est offert au réseau est particulièrement délicate. En effet, les dispositifs de correction d'erreur et de perte génèrent la retransmission de paquets supplémentaires et les mécanismes de contrôle de flux (TCP notamment) régulent les débits de transmission. Les analyses de trafic doivent donc se contenter des données de trafic effectivement mesurées sur des liens, compte tenu de ces retransmissions et régulations.

Le caractère auto-similaire du trafic TCP a été largement étudié dans les articles [5] et [9] ainsi que dans [10]. En complément de ce qui a été dit au paragraphe précédent, notons les tentatives d'explication avancées : aux échelles de temps supérieures à un délai de transmission typique (RTT, de l'ordre de 100 ms), le comportement auto-similaire serait dû à l'extrême variabilité de la taille des documents transférés (la loi de distribution est de type « heavy-tailed », telle la loi de Pareto) ; tandis que les caractéristiques multi-fractales aux échelles de temps inférieures seraient provoquées par les mécanismes de contrôle de congestion du protocole TCP. C'est aussi la conclusion à laquelle [10] est arrivé lorsqu'il a analysé le trafic HTTP (dominant à cette époque dans le trafic Internet). De la même manière, l'article [5] a montré que le trafic Internet peut être représenté par un processus ON/OFF dont la distribution des durées des périodes ON est à queue lourde.

CHAPITRE II

OBJECTIFS ET CONTRIBUTIONS

Le travail décrit dans ce mémoire se propose donc de définir une nouvelle méthodologie pour l'ingénierie des réseaux qui se base sur le mécanisme de prédiction du trafic et de trouver la bonne manière d'exploitation de la prédiction pour une meilleure affectation des ressources dans le réseau.

En effet, pour compléter et affiner la caractérisation du trafic introduite par les premières études et recherches, il est utile de modéliser de façon nouvelle le trafic Internet. L'objectif avoué est de donner un modèle réaliste des arrivées des flux, des paquets et des pertes. Cette action est indispensable, car les informations sur le trafic donneront les informations nécessaires pour la conception, le dimensionnement, la gestion et l'opération d'un réseau. D'autre part, elles donneront aussi les tendances d'évolution du réseau et de ces mécanismes, et permettront de concevoir des simulateurs permettant de confronter les nouveaux protocoles de la recherche à des trafics Internet réalistes. Jusqu'à présent, les ingénieurs et chercheurs en réseau utilisaient le modèle Poissonien pour modéliser les processus d'arrivée de trafic et le modèle de Gilbert pour modéliser les pertes. Pour ce, il faudrait proposer de nouveaux modèles de trafic réalistes. On peut notamment citer les travaux de Padhye [30] qui propose de modéliser le trafic réseau de façon plus réaliste en intégrant les notions d'auto-similarité et de dépendance à long terme LRD.

L'une des tâches essentielles alors, c'est de faire une analyse du trafic capturé, qui a pour objectif de bien définir, voire même de permettre de trouver un modèle pour le trafic Internet, dans notre cas, il s'agit bien d'un trafic auto – similaire qui décrit le plus précisément possible les contraintes réelles liées au trafic et à son environnement. On aborde dans notre sujet les problèmes de caractérisation du trafic Internet. L'objectif est d'en déterminer les caractéristiques intéressantes (à défaut de les déterminer toutes) afin de pouvoir réfléchir à des solutions pour améliorer l'Internet, qui tiennent compte des contraintes liées au trafic et au comportement des utilisateurs, des équipements et des protocoles. On s'intéresse à une certaine classe de modèles paramétriques de série chronologique : les processus longue mémoire généralisés, introduits dans la littérature statistique au début des années 1990. Ces processus à longues mémoires généralisés prennent en compte simultanément dans la modélisation de la série, une dépendance de long terme et une composante cyclique périodique persistante, ce phénomène décrit bien les processus auto-similaires. Ce type de phénomène est fréquent dans de nombreux champs d'application des statistiques, tels que l'économie, la finance, l'environnement ou les transports publics ainsi que dans la téléinformatique, le trafic Internet: c'est la première contribution de notre sujet.

La seconde contribution, décrite dans la partie V, consiste à utiliser les résultats des simulations de trafic Internet plus réalistes afin d'appliquer des algorithmes de prédiction sur ces séries simulées. Quatre algorithmes ont été testés dans ce cadre intégrant des modèles mathématiques afin d'offrir des preuves et des validations du bon fonctionnement des solutions proposées, pour en sortir à la fin, le meilleur algorithme qui permet de donner un taux d'erreurs minimum ainsi qu'une grande simplicité dans son application.

Nous avons également comparé les performances de chacun de ces algorithmes en prévision sur des données qui représentent un trafic réel. Nous avons fourni aussi les expressions analytiques de ces prédicteurs qui pourraient être utilisés en prévision pour des processus à mémoire longue généralisés ainsi que pour d'autres processus à mémoire courte.

Des méthodes ont été utilisées [29], pour ne pas avoir de congestion sur un lien de réseau comme la classification de trafic, ou encore la manière de choisir le routage avec un choix idéale des routes que doivent prendre les paquets en cas de congestion sur un chemin donné sur le réseau, mais on ne s'intéressera pas à ces deux solutions à cause de leurs complexité surtout lors d'un trafic à haute vitesse ou pour une grande quantité de trafic sur le réseau. Notre objectif principal est de fournir une meilleure prédiction de trafic réseau pour bien connaître la bonne bande passante à affecter sur un lien pour ne pas avoir de congestion.

Dans notre cas, on doit commencer par faire une étude sur les caractéristiques d'un trafic auto-similaire et la façon que les surcharges et les congestions se produisent pour ce genre de trafic puis explorer les manières et les modèles idéals qui devraient être utilisées pour réduire la surcharge sur un lien quelconque du réseau.

Et pour que nous puissions comprendre les caractéristiques d'un trafic, et faire une bonne prédiction dans le réseau, on doit utiliser des modèles analytiques, et des modèles statistiques qui doivent être exactes et simples afin de fournir la bonne information sur le trafic et ça nous permettra de faire l'estimation idéale de la variation de trafic sur une petite échelle de temps et puis grandir cette estimation sur une échelle de temps plus grande et développer des modèles pour des prédictions à long terme.

CHAPITRE III

MODÉLISATION DE TRAFIC

3.1 Le trafic Internet

Les réseaux de communication (téléphonie, Internet, réseaux étendus, réseaux locaux, etc.) ont connu, au cours des dernières années, un développement extraordinaire.

Pour leurs opérateurs, une question centrale est de savoir contrôler les flux d'information de façon optimale, afin d'éviter tout problème de congestion ainsi que des problèmes matériels et d'offrir aux utilisateurs un service de bonne qualité, fiable et rapide. Or pour concevoir des procédures efficaces de contrôle de la circulation des informations, pour bien spécifier les équipements matériels nécessaires, une connaissance bien approfondie des propriétés du trafic des communications dans de tels réseaux s'impose.

Mais les réseaux de communication d'aujourd'hui ne sont plus ceux d'hier. Internet a connu une expansion phénoménale ces cinq dernières années (on estime que le trafic de communications vocales qui représentait 90 % du trafic global en 1997, puis représentait 50 % en 2000, n'en représentera que 10 % d'ici 2008). Cet essor a radicalement changé une situation qui était stable depuis plus d'un demi-siècle.

Les raisons profondes de ce développement rapide résident dans l'utilisation, pour l'acheminement de l'information et le contrôle du trafic, de nouveaux protocoles de routage (routage IP, pour *Internet Protocol*) et de contrôle (TCP, pour *Transmission Control Protocol*) décentralisés, qui rendent le réseau Internet indéfiniment extensible.

L'Internet est devenu un réseau multi-service quasi universel sur lequel cohabitent de très nombreux utilisateurs aux usages différents, et de nombreuses applications ayant des besoins différents.

En particulier, les premières études du trafic montrent que le trafic ne possède pas les propriétés de régularité que l'on pensait (modèles de Poisson, de Gilbert, de Markov, etc.), mais bien au contraire, le trafic est particulièrement instable, du à des propriétés de dépendance à long terme particulièrement néfastes à la QoS (Qualité de Service) des services de communication Internet.

3.2 Modélisation de trafic

3.2.1 Modélisation mathématique de trafic

Les modèles du trafic dans le réseau sont basés sur des modèles mathématiques stochastiques utilisant dans le passé les propriétés du modèle Markovien, qui sont des modèles classiques à mémoire courte avec un taux d'arrivée de distribution Poissonienne et une taille de données exponentielle.

Ces anciens modèles ont été utilisés dans l'analyse et l'étude des premiers réseaux ARPANET crée par ARPA (Advanced Research Projects Agency) et le réseau téléphonique.

L'analyse mathématique du trafic dans les réseaux de communication remonte à 1917, avec les travaux engagés par l'ingénieur danois Agner K. Erlang. Puis sa démarche, poursuivie par beaucoup d'autres chercheurs, a fourni les principaux outils mathématiques de dimensionnement utilisés par les opérateurs et les constructeurs de réseaux, jusqu'aux années 1990 environ.

3.2.1.1 Approche Markovienne - Poissienne

Dans ses principes, la démarche mathématique explorée par Erlang et par les autres chercheurs et ingénieurs après lui est *markovienne*. Cela signifie qu'elle décrit le trafic en s'appuyant sur un modèle simple de processus aléatoires qui sont les *chaînes de Markov*, pour lesquelles la théorie mathématique est bien avancée et puissante

3.2.1.2 Définition

Une chaîne de Markov est une suite d'événements aléatoires, dans laquelle la probabilité d'un événement donné ne dépend que de l'événement qui précède immédiatement.

Un processus de Markov est un processus dont l'évolution future $\{ X_s : s < t \}$ ne dépend de son passé qu'à travers son état à l'instant t.

$$\forall s > t, P(X_s \mid X_t : r \leq t) = P(X_s \mid X_t)$$

Cette définition signifie que, pour le future, l'histoire du processus jusqu'à l'instant t est entièrement résumée par son état à l'instant t, ou encore que le présent étant connu, le future est indépendant du passé.

3.2.1.3 Approche

Dans le cadre des réseaux de communication, la démarche markovienne d'Erlang suppose que les lois statistiques caractérisant le trafic sont des lois de Poisson; la loi de Poisson est une des lois de probabilité ou de statistique les plus répandues et les plus simples, elle tire son nom du mathématicien français Denis Poisson (1781-1840). L'hypothèse Poissonienne s'avérait justifiée pour le trafic téléphonique (où les événements aléatoires sont les appels des abonnés, qui surviennent à des instants aléatoires et dont la durée est également aléatoire).

Ce type de modélisation du trafic a permis de mettre en place des procédures de contrôle adaptées pour le réseau. Jusqu'à une date récente, le contrôle des réseaux de communication était un contrôle d'admission, c'est-à-dire que l'opérateur refuse à

l'utilisateur l'accès au réseau lorsque ce dernier ne peut garantir une qualité de service prédéfinie. Ce type de contrôle exige une connaissance assez précise de l'état du réseau dans son ensemble, et il n'est donc possible que pour des réseaux gérés de manière centralisée.

Des modifications et des changements ont eu des conséquences sur le trafic et ses propriétés statistiques, et il a fallu développer une théorie.

En effet, des analyses statistiques effectuées ont montré, d'abord sur des réseaux locaux puis sur le Web, que le trafic ne pouvait plus être décrit à l'aide de lois de probabilité de Poisson. Notamment, on observe des processus aléatoires à *mémoire longue* (où la probabilité d'un événement dépend aussi d'événements qui se sont produits relativement loin dans le passé), ce qui exclut toute modélisation usuelle fondée sur les processus markoviens classiques. Souvent, ces processus présentent également des propriétés statistiques connues sous le nom de multifractalité, qui présentent une très grande irrégularité. Or toutes ces propriétés statistiques ont des conséquences importantes, par exemple pour le dimensionnement des mémoires des routeurs, ne pas en tenir compte, pourraient conduire à sous-estimer les pertes de paquets d'informations par le réseau et entraîner des dysfonctionnements.

Depuis les premiers articles mettant en évidence les nouvelles propriétés statistiques du trafic de données, de très nombreux travaux ont été publiés en vue de les expliquer.

Aujourd'hui, on comprend assez bien l'origine du phénomène de mémoire longue constaté dans la statistique du trafic. On a pu établir qu'il découle directement de la répartition statistique des tailles de fichiers contenus dans les serveurs Web et FTP (protocole de transfert de fichiers) ainsi que des tailles des fichiers demandés par les utilisateurs lors des requêtes HTTP (protocole de transfert hypertexte, utilisé lorsqu'on surfe sur le Web) et FTP. Leurs courbes statistiques, c'est-à-dire les courbes représentant le nombre de fichiers échangés ou consultés en fonction de la taille, décroissent, pour les grandes valeurs, moins rapidement qu'une exponentielle, de part et d'autre de leur maximum: on dit que leur loi de probabilité est *sous - exponentielle*.

Ce que l'on a montré, c'est que les lois statistiques sous- exponentielles auxquelles obéit le comportement individuel des internautes, superposées en grand nombre étant donnée la multitude de ces internautes, ont pour conséquence directe le phénomène de mémoire longue caractérisant le trafic global.

Ces dernières années, un grand nombre de modèles plus ou moins simplifiés ont ainsi été proposés et qui dépassent l'approche traditionnelle, et ce dans des domaines comme les statistiques, la théorie des probabilités et des files d'attente, le contrôle adaptatif de systèmes non linéaires, etc. Certains de ces modèles permettent de rendre compte de la multifractalité du trafic global, propriété évoquée plus haut, d'autres permettent d'évaluer si le partage d'un même canal de communication entre plusieurs flux de données contrôlés par TCP est équitable, etc. Les recherches actuelles se concentrent aussi beaucoup sur l'analyse de DiffServ, une méthode de différenciation des services offerts, fondée sur la création de classes de priorité pour les échanges de données. Cela paraît être la seule démarche extensible capable d'améliorer la qualité de service dans le réseau Internet. Un autre axe important concerne l'adaptation d'UDP (User Datagram Protocol), un protocole utilisé pour les flux de données vidéo et vocales, flux qui ne sont pas régulés par TCP, notamment dans le but de définir des modes de transmission de ces flux qui soient compatibles avec TCP.

Face à ces questions qui présentent des défis scientifiques et des enjeux économiques de première importance, le monde académique et le monde industriel s'organisent. Comment? La plupart des grands groupes industriels des technologies de l'information et des opérateurs ont constitué des équipes de recherche du plus haut niveau, centrées sur la modélisation du trafic et du contrôle dans les réseaux de données, et tout particulièrement dans le réseau Internet. L'effort du monde académique n'est pas moindre, notamment aux États-Unis, en Europe et dans certains pays asiatiques, où se mettent en place des collaborations interdisciplinaires entre des mathématiciens et des chercheurs en informatique ou en génie électrique.

3.2.2 Processus stationnaires

On s'intéressera aux processus stochastiques à temps discret et plus particulièrement aux processus stochastiques stationnaires au sens large, c'est à dire aux processus à covariance stationnaire.

En effet, un processus stochastique $X = \{X_t\}$ est à covariance stationnaire si la moyenne et la variance existent et elles sont indépendantes du temps et aussi si l'auto-covariance est indépendante par translation dans le temps :

1. $E(X_t) = \mu$
2. $E((X_t - \mu)^2) = \sigma^2$
3. $E((X_t - \mu)(X_{t+k} - \mu)) = \gamma$

où μ représente l'espérance non conditionnelle du processus, σ est l'écart type du processus au temps t, et γ représente la fonction d'auto - covariance de retard k.

La fonction d'auto - corrélation est donnée par :

$$\rho(k) = \frac{\mathrm{cov}(X_t, X_{t+k})}{\sqrt{\mathrm{var}(X_t)\mathrm{var}(X_{t+k})}}$$

Étant donné que notre processus est à covariance stationnaire, alors la fonction d'auto - corrélation ρ est sous la forme :

$$\rho(k) = \frac{\gamma(k)}{\sigma}$$

En plus, un processus stationnaire admet un spectre continu avec une fonction de densité spectrale de puissance $S(w)$ pour tout $w \in [-\Pi, \Pi]$, qui n'est autre que la série de fourrier de la fonction d'auto - corrélation:

$$S(w) = \frac{1}{2\pi} \sum_{-\infty}^{\infty} e^{-jwk} \rho(k)$$

3.2.3 Processus à mémoire longue (LRD)

Une définition mathématique peut être donnée pour les processus qui sont caractérisés par leur densité spectrale. Un processus X_t est considéré comme étant un processus longue mémoire, si :

$$S_X(\omega) \approx b \mid \omega \mid^{1-2H}$$

avec $S_X(\omega)$: densité spectrale de X_t, et b : une constante.

Depuis les travaux en hydrologie de Hurst (1951) pour l'analyse des séries de temps chronologiques, le phénomène de la dépendance à long terme ou de longue mémoire s'est élargi à de nombreux autres champs d'application en statistique.

Étant donné un processus stochastique stationnaire $X = \{X_t\}$, on introduit le processus des moyennes issu de X défini tel que :

$$X_t^{(m)} = \frac{(X_{(t-1)m+1} + \ldots + X_{tm})}{m}$$

Cette nouvelle série est très utile pour décrire les propriétés des processus à mémoire longue.

En plus, pour tout m, elle est stationnaire, avec une fonction d'auto-covariance $\gamma^{(m)}$, et de variance $var(X^{(m)})$ et fonction d'auto-corrélation $\rho(X^{(m)})$. Sa variance peut être exprimée en fonction de $var(X)$ et $\gamma(k)$ comme suit :

$$Var(X^{(m)}) = E[(X^{(m)})^2] - E[X^{(m)}]^2$$

$$= \frac{\text{var}(X)}{m} + \frac{2}{m^2} \sum_{k=1}^{m} (m-k)\gamma(k)$$

3.2.3.1 Définition

Pour qu'un processus stochastique $X = \{X_t\}$ (à covariance) stationnaire soit à mémoire longue, il doit vérifier les propriétés suivantes :

1. $\sum_{k=1}^{\infty} \rho(k) = \infty$

2. la densité spectrale est singulière à l'origine,

Les deux propriétés sont équivalentes.

3.2.3.2 Remarque

Ces propriétés ne sont pas vérifiées par les processus de Markov (que ce soit pour le cas du processus de Poisson ou le processus de Poisson doublement stochastique). En plus, les processus de Markov, qui sont des processus à mémoire courte, vérifient les propriétés suivantes :

1. $\sum_{k=1}^{\infty} \rho(k) < \infty$

2. la densité spectrale est finie à l'origine,

3. $m.var(X^{(m)}) \rightarrow \infty$ si $m \rightarrow \infty$

3.2.3.3 Propriété

En pratique, on adoptera une définition moins restrictive. En effet, Un processus stationnaire est dit à mémoire longue tout processus qui vérifie les propriétés suivantes :

1. $\lim_{k \to \infty} \rho(k) \sim C_1 k^{-\alpha}$: (la fonction d'auto corrélation décroît comme une fonction puissance de k)

2. $\lim_{w \to 0} S(w) \sim C_2 w^{-(1-\alpha)}$

3. $\lim_{m \to 0} m.var(X^{(m)}) \sim C_3^{m-\alpha}$

Ce processus est alors à mémoire longue de paramètre $\alpha \in]0,1[$

3.2.4 Différences entre processus à longue mémoire et processus à courte mémoire

Lorsqu'on cherche à modéliser une série chronologique, il arrive que l'on soit confronté au phénomène de dépendance à long terme (longue mémoire) entre les observations de cette série ou encore à un phénomène de dépendance à court terme.

Quelques propriétés qui montrent la différence entre ces deux notions :

3.2.4.1 LRD (Long Range Dependence)

Un processus stationnaire est dit à mémoire longue s'il possède une fonction d'auto-corrélation qui décroît d'une manière hyperbolique, ainsi qu'il doit vérifier les propriétés suivantes :

(i) La fonction d'auto-covariance est hyperboliquement décroissante :

$$\lim_{k \to \infty} \rho(k) \sim C_1 k^{-\alpha}$$

(ii) La somme des auto-covariances est divergente :

$$\sum_{k=1}^{\infty} \rho(k) = \infty$$

(iii) $E[X^{(m)}]$ ne possède pas un bruit de second ordre quand m tend vers ∞

(iv) $Var[X^{(m)}]$ est asymptotiquement de la forme $m^{-\beta}$ pour m grand

(v) $\sum_k Cov(X_n, X_{n+k})$ diverge

(vi) La densité spectrale tend vers $C_2 w^{-(1-\alpha)}$ quand w tend vers 0.

3.2.4.2 SRD (Short Range Dependence)

Un processus stationnaire est dit à mémoire courte s'il possède une fonction d'auto-corrélation qui décroît d'une manière exponentielle. Ce processus doit vérifier les propriétés suivantes :

(i) La fonction d'auto-covariance est exponentiellement décroissante :

$$\lim_{k->\infty} \rho(k) \sim a^{|k|} \quad \text{où} \quad 0 < a < 1$$

(ii) $\sum_{k=1}^{\infty} \rho(k)$ est finie.

(iii) $E(X^{(m)})$ ne possède pas un bruit de second ordre quand m tend vers ∞

(iv) $Var(X^{(m)})$ est asymptotiquement de la forme $Var(X)/m$ pour un m grand

(v) $\sum_{k} Cov(X_n, X_{n+k})$ est convergente.

(vi) La densité spectrale tend vers une constante quand w tend vers 0.

CHAPITRE IV

LE MODELE AUTO-SIMILAIRE

4.1 Introduction

Les processus à mémoire longue et auto–similaire ont été étudiés au milieu du dernier siècle. Ils ont été découverts expérimentalement puis introduits mathématiquement dans de nombreux domaines de la science, comme l'économie et les statistiques et la finance. Au cours de ces dernières années, ces processus ont été utilisés pour modéliser le trafic dans les réseaux de communication moderne que ce soit dans les réseaux locaux Ethernet, les réseaux étendus ou encore dans les réseaux métropolitains. Ce choix a été motivé à la suite des observations statistiques réalisées sur le trafic réel.

Dans ce document, les notions d'autosimilarité, de dépendance à long terme ou encore de caractère fractal sont considérées comme étant des concepts voisins. Nous sommes conscients qu'il existe des différences mathématiques entre elles mais le développement de celles-ci dans le présent rapport n'apporterait pas d'éclaircissements supplémentaires aux problèmes que nous souhaitons aborder. Cependant, on pourra consulter le livre de Beran [12] pour une définition plus approfondie de ces différentes notions. Dans ce papier, on apprend que la notion d'autosimilarité implique la notion de dépendance à long terme. Ainsi, les processus auto - similaire sont considérés comme des cas particuliers de processus comportant de la LRD.

4.2 Les causes possibles de l'autosimilarité

De nombreuses recherches ont été menées depuis quelques années pour déterminer les causes possibles de l'autosimilarité du trafic. Voici les principales conclusions auxquelles elles ont abouti :

4.2.1 Les caractéristiques de trafic (Web)

Les travaux menés par Crovella [10] ont montré que le trafic Web présente des propriétés d'autosimilarité. Ils font apparaître que cette propriété du trafic est essentiellement due à la présence de « queue lourde » dans la distribution des tailles des documents Web. C'est à dire à une présence assez importante de transfert de gros fichiers. A noter qu'il est difficile d'agir sur ce facteur étant donné que la taille des documents ne peut pas être fixée de façon globale pour tout le trafic Internet.

4.2.2 Le comportement Utilisateurs / Applications

Le comportement des utilisateurs et des applications qu'ils utilisent est une des causes possibles de l'autosimilarité. Prenons, par exemple, l'une des applications réseaux les plus utilisées, à savoir le courrier électronique. Lorsqu'on reçoit un courriel, il nous arrive souvent de répondre immédiatement. Il existe ainsi de la dépendance entre les messages envoyés. Ce phénomène induit par le comportement des utilisateurs est encore plus évident dans des applications de « chat ».

Le fonctionnement du protocole HTTP 50 1.0 apparaît également comme une des multiples causes possibles : en effet, lorsqu'on navigue sur le Web, ce protocole ouvre autant de connexions que d'objets dans la page visitée : il crée ainsi de la dépendance entre chaque connexion ouverte pour télécharger chaque élément. Tous ces comportements applicatifs ou humains qui causent de la dépendance à bas niveau sont des facteurs importants créant de l'autosimilarité dans le trafic.

4.2.3 Le mécanisme de contrôle de congestion

De nombreuses études ont montré que les mécanismes de TCP engendrent des modèles de trafic complexes, auto-similaires ou même des comportements chaotiques [13] et [31].

En effet, les mécanismes de congestion de TCP (Slow-Start et Congestion Avoidance) sont tels qu'ils créent une forte dépendance entre les paquets et les pertes de paquets, cette dépendance se retrouve dans l'autosimilarité du trafic.

4.2.4 Les politiques régissant les routeurs

Les politiques d'ordonnancement (scheduling) et de pertes dans les files d'attente (discarding), par la dépendance qu'elles peuvent créer entre les paquets au niveau des files et entre les pertes sont une des causes principales de l'autosimilarité. Par exemple, il a été montré [31] que certaines politiques de discarding, RED51 en particulier, pouvaient permettre dans certaines conditions de réduire la corrélation entre les pertes et donc la LRD par rapport à la politique classique DropTail. Il est également établi que la traversée successive de routeurs ou l'agrégation de trafics au niveau de ces derniers jouent un rôle dans ce phénomène. La liste ci-dessus des causes possibles de l'autosimilarité du trafic Internet n'est pas exhaustive et il existe bien entendu d'autres causes, et peut-être des causes inconnues à l'heure actuelle. Cependant, en l'état actuel des recherches, ce sont les facteurs cités qui apparaissent prédominants dans le phénomène d'autosimilarité.

Dans le but de se familiariser aux caractéristiques des processus auto - similaire, il sera question, dans ce qui suit, de passer sur quelques notions importantes.

4.3 Processus Auto – Similaire

4.3.1 Définitions

Un processus $X = \{X_t\}$ est auto - similaire de paramètre $H \in]0,1[$ si le processus $X^{(m)}$ défini par $X^{(m)} = \frac{(X_{(t-1)m+1} + ... + X_{tm})}{m^H}$ a la même distribution que le processus X pour tout m.

Le passage de X à $X^{(m)}$ correspond à un changement d'échelle. Un processus auto-similaire a donc la même distribution, quelle que soit l'échelle de temps considéré (modulo un coefficient dépendant de l'échelle et du paramètre d'autosimilarité H). Le paramètre H est appelé paramètre de Hurst.

En plus, un processus $X = \{X_t\}$ est exactement auto-similaire au second ordre de paramètre $H \in]0,1[$ si le processus $X^{(m)}$ a la même fonction d'auto-corrélation que X. Ce qui nous permet de dire que :

$$\rho^{(m)}(k) = \tfrac{1}{2}\,\delta^2(k^{2H})$$

$$\text{et } var(X^{(m)}) = \sigma^2 m^{2H-2}$$

où δ est l'opérateur de différence centrale appliqué à une fonction :

$$\delta^2(f(k)) = f(k+1) - 2f(k) + f(k-1)$$

Si on utilise l'équivalence asymptotique de δ et de l'opérateur de dérivation, alors un processus exactement auto - similaire vérifie, quelque soit m :

$$\lim_{k \to \infty} \rho(X^{(m)}) \sim H(2H-1)m^{-(2-2H)},$$

Par conséquent, un processus exactement auto -similaire possède la propriété de mémoire longue si $H > \tfrac{1}{2}$ (en prenant $\alpha = 2-2H$).

4.3.2 Estimation de paramètre d'auto – similarité

Comme on l'avait précisé précédemment, un trafic est dit auto - similaire s'il présente une dépendance de longue mémoire (LRD).

Pour vérifier cette propriété, il est possible d'estimer le paramètre de Hurst *H*. Si ce paramètre est strictement supérieur à 0.5, alors on pourrait dire que le trafic montre une mémoire longue ou une dépendance à long terme; si la valeur estimée est proche 0.5 alors on serait dans le cas d'un trafic non auto - similaire et ayant une mémoire courte.

De nombreuses méthodes qui permettent d'estimer ce paramètre à partir d'une réalisation $\{X1, \ldots, X_n\}$, parmi elles on cite :

4.3.2.1 Méthode statistique R/S

Pour un processus $X = \{X_t\}$, on définit la méthode « Rescaled Adjusted Range » :

$$R(n) = max\ (0, W_1, \ldots, W_n) - min\ (0, W_1, \ldots, W_n) \qquad \text{Eq-4.3.2.1}$$

avec $W_k = (X_1 + \ldots + X_k) - k\overline{X}\ (n)$, $E(X(n))$ est l'espérance du processus,

et $S(n)$ est sa variance.

- Si $E\{R\ (n)/S\ (n)\}$ est asymptotiquement proportionnel à $n^{1/2}$ alors ce processus est à mémoire courte (SRD).
- Si, par contre, $E\{R\ (n)/S\ (n)\}$ est proportionnel à n^H, alors ce processus est à longue mémoire (LRD).

Pour estimer la valeur du paramètre de Hurst *H*, pour une série de *N* valeurs, on subdivise notre série en *K* sous ensembles et on estime alors $R(t_i,n)/S(t_i,n)$

Pour tout $t_i = iN/K + 1$ où i est tel que $(t_i - 1) + n < N.R(t_i,n)$ et qui est défini comme en *(Eq-4.3.2.1)* en remplaçant W_k par $W_{ti+k} - Wt_i$ et $S(t_i, n)$ par la variance empirique de $\{X_{ti}, \ldots, X_{ti+n}\}$.

Ce qui permet d'avoir plusieurs valeurs de *R/S* pour chaque valeur de *n*. L'estimation de *H* se fait en traçant $log(R(t_i,n)/S(t_i,n))$ en fonction de $log(n)$, la droite obtenue aura une pente de valeur *H*.

4.3.2.2 Méthode graphique de variance

Pour un processus à mémoire longue (LRD) de paramètre α, la variance du processus des moyennes est asymptotiquement équivalente à $m^{-\alpha}$.

Cette méthode consiste à tracer la courbe $log\ (var(X^{(m)}))$ en fonction de logarithme de *(m)* qui est une droite de pente $-\alpha$ pour *m* assez grand. Il suffit de faire une régression linéaire pour obtenir une estimation de α, et par la suite de paramètre de Hurst *H*.

$H = 1 - \alpha/2$ pour un processus exactement auto – similaire.

$H = (3 - \alpha)/2$ pour un processus auto – similaire.

4.4 Conclusion

En fait, il est aujourd'hui établi que le trafic Internet n'est pas Poissonnien, mais qu'il possède plutôt des propriétés de dépendance à long terme et d'autosimilarité. Les conséquences de cette découverte sont d'une importance capitale. En effet, les processus auto-similaires, par rapport aux processus Poissoniens, présentent la caractéristique d'avoir une variance très importante. Qualitativement, cela signifie qu'il est indispensable de surdimensionner les liens et les tailles des files d'attente dans les routeurs pour prendre en compte cette variance. Quantitativement, l'évaluation du facteur de Hurst d'un processus auto-similaire permettra de quantifier le niveau de

surdimensionnement nécessaire pour un fonctionnement optimal du réseau. Comme nous l'avons expliqué au début de ce chapitre, la modélisation complète du trafic réseau est à l'heure actuelle une tâche très difficile.

Nous avons donc travaillé sur des processus auto-similaire qui montre largement les caractéristiques d'un trafic réseau réel. On s'est intéressé à la prédiction des séries chronologiques utilisant un processus à longue mémoire. L'ensemble de la contribution réalisée est présenté dans le chapitre qui suit.

CHAPITRE V

PREDICTION DE TRAFIC

5.1 Objectif

Le but de la prédiction de trafic dans le réseau est de prévoir les observations futures sur la quantité de trafic. Il y a une grande dépendance entre les futurs observation X_{n+k} et les valeurs prises du passé X_n , X_{n-1} , X_{n-2} , ... X_0. Et pour obtenir une bonne prédiction, il est important d'utiliser le modèle approprié, le plus efficace et qui donne un taux d'erreur le plus bas possible, et par la suite, ça nous permettra une implémentation facile avec un minimum de paramètres à estimer.

Pour des processus à longue mémoire, une bonne prédiction à long et à court terme peut être obtenue avec l'existence d'un grand nombre d'enregistrements pris du passé.

Pour avoir une meilleure prédiction, il faut se baser sur ces critères [3]:

- bien choisir le modèle de prédiction pour fournir une grande exactitude ;
- afin d'achever une prévision à temps réel, il faut avoir une certaine simplicité dans le choix des paramètres à utiliser et à implémenter ;
- la majorité des modèles de modélisation de trafic ont été fait avec des données offline, on voudrait faire alors cette prédiction avec des données et des valeurs prises on-line ;
- un bon modèle de prédiction qui doit s'adapter à tout changement de trafic.

Différents outils ont été proposés pour effectuer les prédictions sur les processus à longue mémoire (*LRD*) et auto - similaire. Les principaux modèles sont :

- Les processus *ARIMA*, par leurs simplicité et flexibilité, ces modèles sont devenus très populaire dans leur application dans l'analyse des séries de temps (séries chronologiques).
- Les processus *FBM* ou Fractional Brownian Motion. Ce sont des processus dérivés du précédent processus (*FGN*), qui sont aussi en temps continu. Les processus *FGN* ou Fractional Gaussien Noise (bruit gaussien fractionnaire). Ce sont des processus en temps continu, et on peut utiliser ce modèle pour effectuer des prévisions sur les séries chronologiques.
- Les processus *F-ARIMA*, ce sont des processus en temps discret. C'est une extension des processus *ARIMA* (les processus *ARIMA* sont un sous-ensemble des processus *F-ARIMA*). Il est possible de faire des prévisions à partir de ces processus.
- *LMMSE* qui dérive de l'hypothèse d'un processus stationnaire stochastique d'espérance zéro. Il permet une très bonne prédiction.

La liste n'est probablement pas exhaustive. Nous avons donc choisi d'expérimenter les méthodes *ARIMA*, *F-ARIMA* et *FBM* ainsi que *LMMSE*, car c'est pour ces processus que nous avons trouvés des méthodes d'estimation abordables et compréhensibles. C'est aussi parce que les processus en temps continu, qui aident souvent à comprendre des sujets complexes, ont un intérêt moindre d'un point de vue pratique

5.2 Algorithmes et méthodes de prédiction

5.2.1 Les modèles *ARMA* et *ARIMA*

En général, les séries chronologiques sont traitées avec les modèles du type *AR*, *MA*, *ARMA*, *ARIMA (AutoRegressive Integrated Moving Average*), notons que les modèles *AR* et *MA* sont des sous modèles de *ARIMA*. Ce sont les modèles usuels, que l'on trouve dans la plupart des logiciels statistiques.

Ces modèles marchent très bien s'il s'agit, par exemple, de prévoir le trafic journalier d'un aéroport ou encore d'une station de transport et, plus généralement, de prévoir des processus à courte mémoire et quelques autres processus à longue mémoire.

Par contre, dans le cas de la bourse, on est confronté à un signal beaucoup plus variable et incertain. Il s'agissait du caractère nécessairement discontinu des prix de la bourse, dont les changements se concentrent dans le temps, du caractère cyclique mais non périodique de l'évolution économique et de diverses conséquences de ces observations sur le calcul des risques.

Depuis, d'autres outils statistiques ont été apportés qui répondent aux caractéristiques observées par Mandelbrot. Les processus longue mémoire (ou processus en 1/F) semblent être adaptés à la prévision de la bourse. Ces processus ont une persistance beaucoup plus forte avec le passée. Les processus *ARIMA* ont une extension dans la famille des processus en 1/F qui leur permet de garder cette persistance sans augmenter le nombre des paramètres à estimer (les processus *FARIMA : Fractional ARIMA* qui sont une extension du modèle classique *ARIMA*). Pour ces raisons, on ne s'étendra pas d'avantage sur les processus à mémoire courte, mais par contre il y aura une recherche documentaire ainsi que des simulations et des travaux sur les processus à longue mémoire dans la section suivante.

Le modèle *ARIMA* a été introduit par Box et Jenkins en 1970. Par leurs simplicité et flexibilité, ces modèles sont devenus très populaire dans leur application dans l'analyse des séries de temps (séries chronologiques).

5.2.1.1 Définitions

Si X_t un processus *AR* (Auto Regressif), cela veut dire que X_t est la somme d'une fonction linéaire de son propre retard (X_{t-1}, X_{t-2} ,... X_{t-q}), et d'un bruit aléatoire, ε_t.

$$X_t = a_1 X_{t-1} + a_2 X_{t-2} + \ldots + a_q X_{t-q} + \varepsilon_t$$

Le modèle *MA* (Moving Average) est un modèle où X_t dépend des chocs aléatoires persistants. (Un choc aléatoire peut influencer plusieurs périodes).

$$X_t = b_1 \varepsilon_{t-1} + b_2 \varepsilon_{t-2} + \ldots + b_p \varepsilon_{t-p}$$

Le modèle *ARIMA* est une combinaison des deux modèles précédents. C'est un modèle plus complet qui prend en compte à la fois son propre retard et des chocs aléatoires persistants.

Pour simplifier, on suppose que $\mu = E(X_t) = 0$. Autrement, X_t doit être remplacé dans toutes les formules par $(X_t - \mu)$.

Dans la suite, B représente l'opérateur retard (Backshift) définie par :

$$BX_t = X_{t-1} \qquad et \qquad B^2X_t = X_{t-2} \ldots$$

En particulier, la différence peut être exprimée comme :

$$X_t - X_{t-1} = (1-B)X_t ,$$

$$\text{et} \quad (X_t - X_{t-1}) - (X_{t-1} - X_{t-2}) = (1-B)^2 X_t , \ldots$$

Soient les entiers p et q définis par :

$$\Phi(x) = 1 - \sum_{j=1}^{p} \Phi_j x^j$$

$$\text{et } \Psi(x) = 1 + \sum_{j=1}^{q} \Psi_j x^j$$

On suppose que les solutions de l'équation $\Phi(x)=0$ et de l'équation $\Psi(x)=0$ n'appartiennent pas à l'intervalle $[0,1]$.

Soit $\{\varepsilon_t\}$ des variable normales identiquement et indépendamment distribuées (iid) et qui forment un processus bruit blanc centré de variance finie σ et d'espérance nulle.

Un modèle $ARMA(p,q)$ est définie par l'équation suivante :

$$\Phi(B)\, X_t = \Psi(B)\, \varepsilon_t \,.$$

Un modèle $ARIMA(p,d,q)$ est définie par :

$$\Phi(B)(1-B)^d X_t = \Psi(B)\, \varepsilon_t. \quad \text{(Eq-5.2.1.1))}$$

Notons qu'un modèle $ARMA(p,q)$ est un processus $ARIMA(p,0,q)$.

Si $p = 0$, on dit que le processus est $MA(q)$

Si $q = 0$, on dit que le processus est $AR(p)$

5.2.1.2 Propriétés [14]

Propriété 1 : Un processus *ARMA* vérifie:

(i) Si le polynôme $\Phi(x)$ ne s'annule pas sur le cercle défini par $|x| = 1$, alors le processus $\{Xt\}$ est un processus stationnaire linéaire

(ii) Si le polynôme $\Phi(x)$ ne s'annule pas sur le cercle défini par $|x| \leq 1$, alors le processus $\{Xt\}$ est un processus qui possède une représentation causale.

(iii) Si le polynôme $\Psi(x)$ ne s'annule pas sur le cercle défini par $|x| \leq 1$, alors le processus $\{Xt\}$ possède une représentation inversible.

Si $d>1$, la série originale $\{Xt\}$ n'est pas stationnaire. Pour obtenir un processus stationnaire, il faut dériver d fois.

On peut simplifier l'équation, lorsque l'entier d est positif alors on pourrait écrire $(1-B)^d$ de la manière suivante :

$$(1-B)^d = \sum_{k=0}^{d} C_k^d (-1)^k B^k$$

avec $$C_k^d = \frac{d!}{k!(d-k)!} = \frac{\Gamma(d+1)}{\Gamma(k+1)\Gamma(d-k+1)}$$

où $\Gamma(x)$ est la fonction gamma tel que : $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t}\, dt$

On généralise maintenant la définition, au cas des processus *ARMA (p,q)* intégrés d'ordre d, ou *ARIMA (p, d, q)*.

Un processus de second ordre $\{X_t\}$ est défini comme étant un processus *ARIMA(p,d,q)*, si le processus $((I-B)^d X_t)$ est un processus *ARMA*.

Le logiciel statistique *S-PLUS* permet de faire une simulation des processus *AR, MA, ARMA* et *ARIMA*, à l'aide de la fonction prédéfinie *arima.sim()*. Cette fonction prend comme arguments un objet de type liste qui contient les valeurs des paramètres à utiliser

Par exemple, pour simuler un processus *ARMA(2,1)* suivant qui vérifie l'équation suivante:

$$(I - 0.5B - 0.2B^2)X_t = (I - 0.4B)\,\varepsilon_t.$$

Le modèle est spécifié puis généré de la manière suivante:

```
Arma21.mod<-list(ar=c(0.5,0.2),ma=0.4)
```

On pourrait simuler dans *S-PLUS* une trajectoire de longueur *500* issue du processus *ARMA(2,1)* qui pourrait représenter en fait, un trafic réel, et ce à l'aide de la commande suivante :

arma21.mod<-list(ar=0.6,ma=0.3)

Xarma21<-arima.sim(n=500,model=arma21.mod)

La fonction *arima.mle()* permet de renvoyer une liste qui contient les valeurs des paramètres estimés, les ordres du modèle, la matrice variance covariance des paramètres et la variance résiduelle estimée, aussi ce logiciel nous permet de savoir si l'algorithme converge ou non.

Et comme résultat, on obtient un objet de type vecteur, et que l'on peut transformer en série chronologique de type *cts* ou *rts*. On obtient finalement cette représentation graphique :

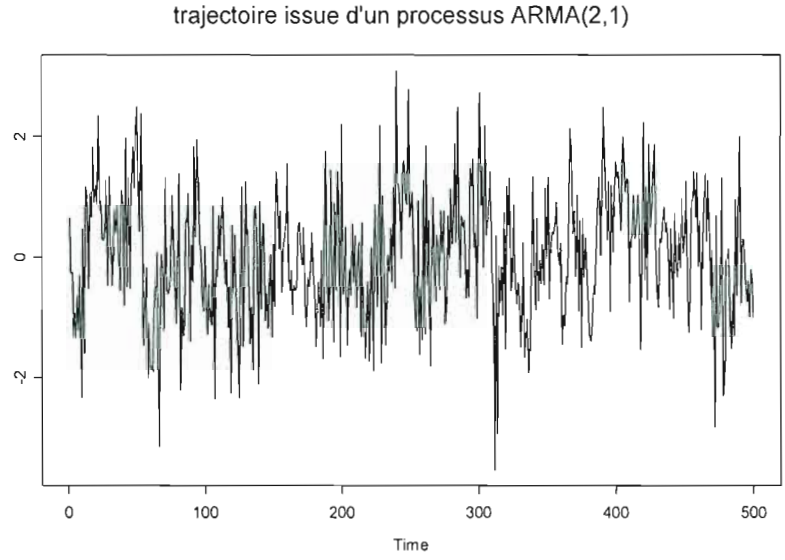


Figure 5.1 Trajectoire d'un processus ARMA(2,1)

Propriété 2 : La densité spectrale d'un processus *ARMA* est donnée par :

$$f_{ARMA}(\lambda) = \frac{\sigma^2}{2\pi} \frac{|\Psi(e^{i\lambda})|^2}{|\Phi(e^{i\lambda})|^2}$$

et la densité spectrale d'un processus *ARIMA* est donnée par :

$$f(\lambda) = |1-e^{i\lambda}|^{-2d} f_{ARMA}(\lambda) \qquad \text{(Eq-5.2.1.2)}$$

avec $|1-e^{i\lambda}| = 2 \sin \frac{1}{2} \lambda$,

Si $\lambda \rightarrow 0$, on aura $2 \sin \frac{1}{2} \lambda \rightarrow \lambda$ d'où $f(\lambda) = |\lambda|^{-2d} f_{ARMA}(0)$

Pour générer une trajectoire de longueur *300* utilisant le logiciel *S-PLUS* issue d'un processus *ARIMA(1,1,0)* de paramètre $\Phi_1 = 0.4$, à l'aide de la commande suivante:

```
>xarima<-arima.sim(n=500,model=list(ar=0.4,ndiff=1))
```

Pour avoir comme résultat, un objet de type vecteur, que l'on peut transformer en série chronologique de type *cts* ou *rts*. Qui aurait la représentation graphique suivante (Cette série générée possède les caractéristiques d'un processus auto-similaire avec une dépendance à long terme, sous la condition $0 \leq d \leq 1/2$):

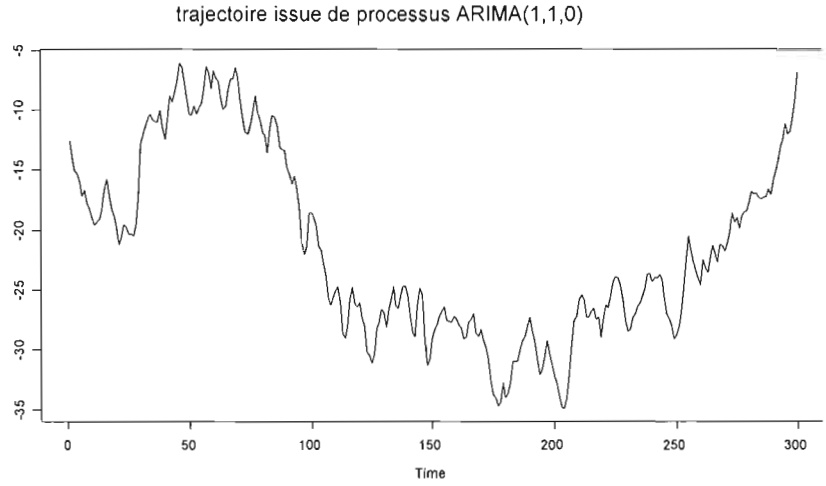


Figure 5.2 Trajectoire d'un processus ARIMA(1,1,0)

5.2.1.3 Remarques

Revenant à l'équation (Eq-5.2.1-(1))

* si $d=0$, alors X_t est un processus $ARMA(p,q)$.

* si $0<d<1/2$, alors X_t a la propriété de LRD

La covariance est donnée par :

$$\gamma(k) = c(k)|k|^{2d-1}$$

avec $$c(k) = \frac{\sigma^2}{\pi} \frac{|\Psi(1)|^2}{|\Phi(1)|^2} \Gamma(1-2d) \sin \Pi d$$

La corrélation est donnée par :

$$\rho(k) = \alpha(k)|k|^{2d-1}$$

avec $\alpha(k) = \dfrac{c(k)}{\int_{-\pi}^{\pi} f(\lambda)\, d\lambda}$

5.2.1.4 Propositions

Appliquant la formule de Gradshteyn et Ryzhik [11],

La covariance pourrait s'écrire, pour un processus *ARIMA(0,d,0)* :

$$\gamma(k) = \sigma^2 \frac{(-1)^k \Gamma(1-2d)}{\Gamma(k-d+1)\Gamma(1-k-d)}$$

La corrélation pourrait s'écrire, pour un processus *ARIMA(0,d,0)*:

$$\rho(k) = \frac{\Gamma(1-d)\Gamma(k+d)}{\Gamma(d)\Gamma(k+1-d)}$$

Le comportement asymptotique de $\rho(k)$ est donnée par :

$$\rho(k) = \frac{\Gamma(1-d)}{\Gamma(d)} |k|^{2d-1} \quad (|k| \to \infty).$$

Soit X_t un processus stationnaire, qui vérifie :

$$\Phi(B)(1-B)^d X_t = \Psi(B)\, \varepsilon_t \qquad \text{(Eq-5.2.1.4)}$$

Si $-1/2<d<1/2$, alors (X_t)est appelé un processus *F-ARIMA(p,d,q)*.

Pour $d>1/2$ le processus n'est plus stationnaire, et pour $0<d<1/2$ le processus est à mémoire longue et a le comportement d'un modèle auto-similaire.

L'équation (Eq-5.2.1.4) pourrait être écrit de la manière suivante :

$$(1-B)^d X_t = \widetilde{X}_t$$

avec $\widetilde{X}_t$ est un processus *ARMA* définie par : $\widetilde{X}_t = \Phi(B)^{-1}\Psi(B)\,\varepsilon_t$.

5.2.2 Prédiction

Après avoir spécifié et estimé les paramètres du processus *ARIMA*, à fin de générer un processus stationnaire, et plus précisément qui possède les caractéristiques d'un processus auto-similaire, on s'intéresse maintenant à l'utilisation du modèle *ARIMA* pour effectuer des prédictions sur cette série chronologique simulée.

On notera $\hat{X}_t(h)$ le prédicteur à l'horizon h (h est un entier positif) de X_{t+h}, pour tout $t \in Z$ et tout $h > 0$.

L'erreur de prédiction est exprimée de cette façon :

$$e_{t+h} = X_{t+h} - \hat{X}_t(h)$$

En particulier, dans le cas d'un processus *ARIMA(p,q)*, le prédicteur à horizon $h = 1$ est donné par :

$$\hat{X}(1) = \phi_1 X_t + \ldots + \phi_p X_{t-p+1} - \theta_1 \hat{\varepsilon}_t - \ldots - \theta_q \hat{\varepsilon}_{t-q+1}$$

Le logiciel *S-PLUS* utilise le processus sous la forme d'un modèle espace – état et utilise un filtre de Kalman [11] pour obtenir les prévisions ainsi que leurs écarts types.

La prévision d'un processus *ARIMA* se fait à l'aide de la fonction prédéfinie *arima.forecast()* qui prend comme argument obligatoire la série d'étude, le nombre d'horizon h et le modèle estimé renvoyé par *arima.mle()&model*. La fonction

arima.forecast() renvoie une liste deux éléments de type vecteur contenant les valeurs de la série prédite(*&mean*) et les valeurs de l'écart type de la série prédite.

Pour un modèle *ARIMA(1,0.4,1)*, où *d=0.4*, ce qui donne la valeur de *0.9* pour le paramètre de Hurst *H*, on pourrait prévoir à partir d'une liste de *100* valeurs, par exemple, les *20* prochaines valeurs (un horizon *h=20*) , et afficher le graphique qui contient le prolongement de la courbe lors de la prédiction.

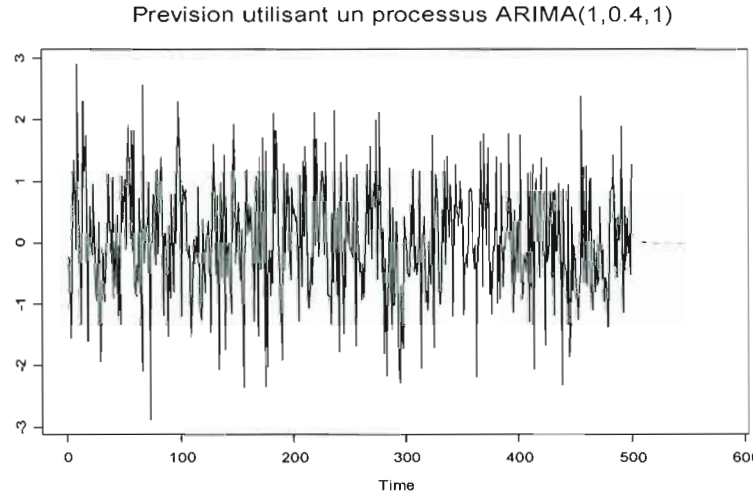


Figure 5.3 Prédiction avec le modèle ARIMA

À partir du graphique résultant des prévisions, on remarque que les prévisions convergent rapidement vers la moyenne non conditionnelle de la série, à savoir zéro dans ce cas. Ce qui caractérise les processus *ARIMA*. En fait, on observe une convergence plus lente des prévisions, de manière hyperbolique, vers la moyenne non conditionnelle peut être obtenue à l'aide des processus à mémoire longue.

Soit le tableau suivant qui permet de nous calculer le taux d'erreurs $e_{t+h} = X_{t+h} - \hat{X}_t(h)$ pour les *20* valeurs prédites par notre modèle de prévision *ARIMA* :

i	X_i	$\hat{X}_i$	$\lvert \hat{X}_i - X_i \rvert$	$\frac{\lvert \hat{X}_i - X_i \rvert}{\lvert X_i \rvert}$
81	2.14	1.06	1.08	50.3 %
82	0.53	0.25	0.28	51.9 %
83	1.52	1.10	0.42	27.9 %
84	0.21	1,01	0.79	36.9 %
85	0.36	0.11	0.25	22.74 %
86	-0.40	0.28	0.12	29.99 %
87	0.30	0.13	0.16	54.62 %
88	1.67	0.91	0.75	45.26 %
89	1.10	1.84	0.74	67.45 %
90	0.93	1.80	0.86	92.88 %
91	-0.11	0.57	0.45	39.33 %
92	-1.54	-1.63	0.09	07.44 %
93	-0.14	-0.19	0.05	37.78 %
94	-0.74	-0.22	0.52	69.97 %
95	0.14	0.44	0,30	21.11 %
96	0.71	0.23	0,48	67.65 %
97	1.63	1.27	0,35	21.87 %
98	0.23	0.10	0,12	54.66 %
99	1.49	1.03	0,46	30.90 %
100	0.45	0.92	0,46	50.91 %

Tableau 5.1 L'erreur et le taux d'erreur de la prédiction - ARIMA

On remarque bien que le taux d'erreur varie d'une valeur à une autre, il est souvent assez grand, souvent il y a une bonne différence entre la valeur prédite et la valeur réelle.

Ces tests ont été faits pour une valeur de paramètre de Hurst H égale à *0.9*, et appliquées sur une série chronologique de taille initiale *80* à fin de prévoir les *20* prochaines valeurs.

Essayons maintenant de faire varier la taille de la série chronologique, tout en laissant invariant H égale à *0.85* (la valeur qui identifie un trafic Internet réel), à fin d'étudier l'effet de la taille de la série chronologique initiale sur le taux d'erreur de prédiction.

5.2.2.1 Simulation de la série avec *ARIMA*

Le taux d'erreur résultant est donné par : $\left|\frac{\hat{x}(n+1)-x(n+1)}{x(n+1)}\right|$

Les tests effectués sur des séries simulées par *ARIMA*, en variant à chaque fois la taille de la série allant de *100* valeurs jusqu'à *30000* valeurs (on ne pouvait pas prendre une taille plus importante parce qu'à partir d'une taille n supérieure à *35000*, le logiciel *S-PLUS* ne répond plus).

Taille	*Le taux d'erreur résultant pour les 10 prochaines valeurs (en %)*									
100	91.0	89.2	99.8	100.1	99.9	99.9	100	99.9	99.9	100
1000	94.9	94.0	16.9	67.06	79.70	91.6	94.1	86.6	76.9	80.3
5000	86.7	2.31	10.7	99.38	56.57	27.6	99.9	100	80.7	95.6
10000	89.3	45.7	91.5	64.37	16.00	15.4	111	92.9	84.7	57.6
20000	92.6	85.2	87.9	7.11	108.9	34.86	94.97	94.62	22.66	85.26
30000	8.9	8.9	56.4	101.6	59.48	22.88	62.00	47.33	36.73	53.63

Tableau 5.2 Le taux d'erreur VS la taille - ARIMA

Taille_série	100	1000	5000	10000	20000	30000
Moyenne_Erreur	98.02	78.22	65.97	66.99	71.42	51.81

Tableau 5.3 Le taux moyen d'erreur en variant la taille

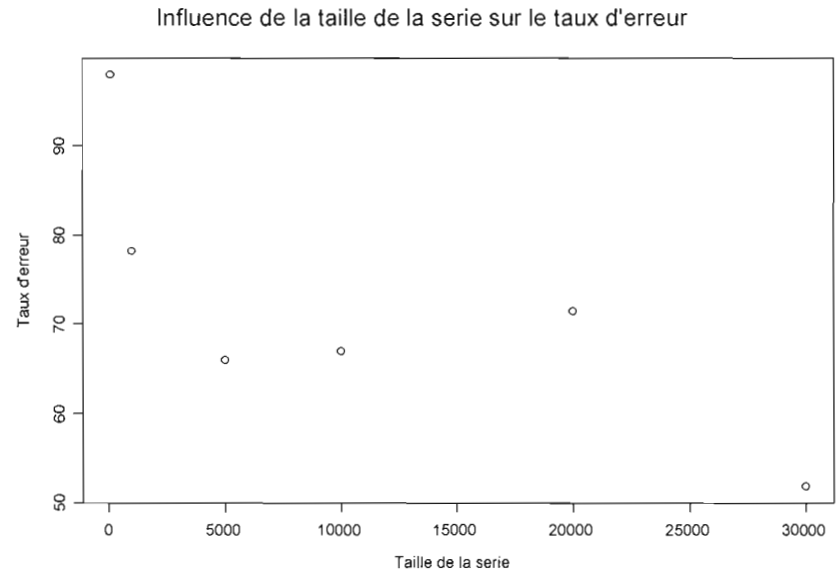


Figure 5.4 Influence de la taille de la série sur le taux d'erreur

On remarque, en étudiant tous les cas, que le taux d'erreurs diminue à chaque fois qu'on augmente la taille de la série, c'est à dire, une liste qui contient le plus grand nombre de valeurs possibles prises du passé, donne des taux d'erreur inférieurs et par la suite une meilleure prédiction.

Pour chaque série, on a effectué la prédiction des 10 prochaines valeurs, avec un paramètre de Hurst H égale à 0.85 qui caractérise bien le comportement d'un trafic Ethernet réel. Et à chaque fois on affiche le taux d'erreur, ainsi que la moyenne des 10 taux d'erreurs. On remarque que cette moyenne diminue lorsque la taille de la liste augmente.

À partir d'une taille de la série de 5000 valeurs, le taux d'erreurs obtenu est acceptable, ce qui nous amène à conclure qu'il faut se baser sur une bonne historique de taille assez énorme pour spécifier le modèle et pour obtenir des bons résultats en prédiction.

5.2.2.2 Les séries Bytes et Packet

Les fichiers Byte et Packet contiennent les valeurs d'une série chronologique d'un trafic Ethernet réel, les valeurs ont été collectionnées à Bellcore en Août *1989* à toutes les *10* millisecondes, avec une taille totale de *360,000* pour chaque fichiers, ces fichiers sont accessible sur le site Web de Murad Taqqu à l'adresse suivante :

http://math.bu.edu/people/murad/methods/time_series/index.html#Ethernet

On va effectuer nos tests sur ces deux séries qui représentent un trafic Internet, les chercheurs Leland, Taqqu, Willinger et Wilson [5] ont montré, comme mentionné dans les paragraphes précédents, que le trafic Ethernet possède les propriétés d'un trafic auto - similaire

En utilisant des essais sur le fichier au complet, de taille *36,000*, l'application se fige à cause de la taille énorme des données, on prendra alors une partie qui représente la partie la plus grande possible de chaque fichier pour effectuer nos tests de prédiction.

Le tableau 3.2 suivant contient les valeurs prédites, huit valeurs pour nos tests, l'erreur de prédiction ainsi que le taux d'erreur la série Byte en utilisant le modèle *ARIMA* en variant à chaque fois les valeurs de d :

d	X_i	$\hat{X}_{i+1}$	$\lvert \hat{X}_{i+1} - X_{i+1} \rvert$	$\frac{\lvert \hat{X}_{i+1} - X_{i+1} \rvert}{\lvert X_{i+1} \rvert}$
	1	0.61	0.38	38.59%
	1	-0.008	1.008	100.83%
	2	0.00	1.99	99.99%
	1	1.65	0.65	65%
0.4	1	1.57	0.57	57.80%
	5	1.00	3.99	23.43%
	4	2.67	1.32	33.04%
	0	-1.76	1.76	***
	1	0.09	0.90	90.38%
	1	-0.05	1.05	105.53%
	2	0.03	1.96	98.40%
	1	-0.01	1.01	101.83%
0.3	1	0.01	0.98	98.94%
	5	-0.006	5.006	100.12%
	4	0.003	3.99	99.91%
	0	-0.002	0.002	***
	1	0.28	0.71	71.89%
	1	-0.09	1.09	109.88%
	2	0.034	1.965	98.26%
	1	-0.012	1.012	101.22%
0.2	1	0.004	0.995	99.57%
	5	-0.001	5.001	100.03%
	4	0.0005	3.9994	99.98%
	0	-0.0001	0.0001	***

0.1	1	0.156	0.843	84.36%
	1	-0.047	1.047	104.72%
	2	0.034	1.993	99.68%
	1	-0.0008	1.0008	100.08%
	1	0.0001	0.9998	99.98%
	5	-0.00	5.00	100.00%
	4	0.000001	3.99999	99.99%
	0	0.002	0.002	***

Tableau 5.4 Le taux d'erreur - La liste Byte

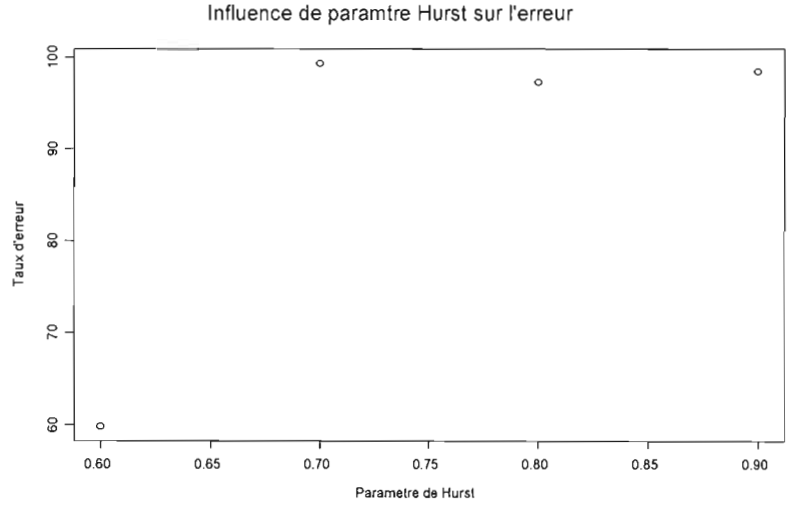


Figure 5.5 Influence de paramètre de Hurst sur l'erreur

On a utilisé une séquence de *1000* premières valeurs prises du fichier Byte puis du fichier Packet, puis on a appliqué le modèle *ARIMA* pour effectuer la prédiction de quelques valeurs de futur, en connaissant les valeurs réelles.

Les huit prochaines valeurs réelles sont :

1	1	2	1	1	5	4	0

À chaque fois, on donne la liste des valeurs prédites, l'erreur ainsi que le taux d'erreurs en variant respectivement les valeurs de paramètres de Hurst H, dans notre cas en variant la valeur de d, avec $d = H - ½$. D'où le tableau 5.4.

On remarque qu'à chaque fois qu'on augmente la valeur de d, et donc la valeur de H, le taux d'erreur serait le meilleur; la meilleure valeur est donnée lorsque H est égale à *0.9*, qui représente dans la majorité des cas un trafic Ethernet réel.

En appliquant le modèle *ARIMA* pour prédire le trafic future à partir d'une liste initiale contenant les valeurs de la liste Packet, les tests ont été faits en comparant les valeurs estimées obtenues par le modèle avec les valeurs, on aboutait au tableau 5.5,

Les huit prochaines valeurs réelles sont :

64	1266	128	64	64	5450	3336	162

D'après les tests effectués, Tableau.5.5 et Tableau.5.6, le modèle *ARIMA* permet de donner de bons résultats pour une prédiction à un horizon h égale à 1 et pour un paramètre de Hurst H égale à *0.9*, ce qui représente les mêmes résultats obtenus sur la série Byte. Figure 5.5.

Mais pour une prédiction d'horizon supérieure à *1*, les taux d'erreurs obtenus sont assez grands.

d	X_i	$\hat{X}_{i+1}$	$\lvert \hat{X}_{i+1} - X_{i+1} \rvert$	$\frac{\lvert \hat{X}_{i+1} - X_{i+1} \rvert}{\lvert X_{i+1} \rvert}$
	64	6.08	2.08	03.25%
	1266	1.22	1267.22	100.09%
	76	02.04	74.040	122.71%
0.4	64	14.53	78.53	80.13%
	5	2.716	51.28	99.85%
	5450	8.003	5441.99	23.43%
	3336	67.46	2468.53	73.99%
	162	81.812	219.812	135.68%
	64	30.847	33.152	107.47%
	1266	-8.305	1274.30	153.43%
	128	100.583	27.416	27.25%
0.2	64	-33.265	97.265	292.39%
	64	19.15	44.84	234.19%
	450	-3.29	453.296	654.36%
	3336	419.103	2916.896	695.98%
	162	76.192	85.807	112.61%

Tableau 5.5 Le taux d'erreur - la liste Packet

Hurst	0.9	0.7
Moyenne_erreur	79.89	284.71

Tableau 5.6 Le taux moyen d'erreur - La liste Packet

5.2.2.3 Simulation de la série avec *FGN*

Appliquant maintenant, le même modèle *ARIMA* sur des séries simulées par un processus *FGN* qui pourraient représenter bien évidement un trafic auto – similaire.

Nos tests ont été faits alors sur des séries simulées par *FGN*, en variant à chaque fois la taille de la série qui pourrait aller de *100* valeurs jusqu'à *30000* valeurs (mais à partir d'une taille supérieure à *35000*, le logiciel *SPLUS* ne répond plus).

Taille	Le taux d'erreur résultant pour les 10 prochaines valeurs (en %)									
100	7.62	37.18	37.72	279.6	44.89	250.1	48.26	50.90	114.7	231.2
1000	319.9	77.70	251.3	43.56	60.49	89.48	64.58	73.31	17.59	9.89
10000	81.76	65.04	78.82	63.87	79.01	64.79	79.67	66.01	80.41	67.23
20000	65.85	35.49	43.33	72.69	75.02	47.31	26.62	50.56	78.04	53.55
30000	46.85	16.87	67.61	10.33	22.02	18.32	23.94	23.25	75.43	27.52

Tableau 5.7 Les 10 prochaines valeurs de l'erreur variant la taille

Taille_Série	100	1000	10000	20000	30000
Moyenne_Erreur	110.22	100.79	72.66	54.84	33.21

Tableau 5.8 Le taux moyen d'erreur en variant la taille

On remarque dans ce cas, pour les séries simulées par *FGN*, que le taux d'erreurs diminue à chaque fois qu'on augmente la taille de la série, c'est à dire, que la prédiction est meilleure si on se base sur une série de grande taille, et d'un grand historique.

Pour chaque série, on a effectué la prédiction des *10* prochaines valeurs, avec un paramètre de Hurst *H* égale à *0.85*, Et à chaque fois on affiche le taux d'erreur, ainsi que la moyenne des *10* taux d'erreurs obtenus, on remarque que cette moyenne diminue quand on se base sur une série qui a une taille la plus grande possible, mais ce modèle reste inapproprié puisque le taux d'erreurs n'est pas souvent le meilleur et le plus petit possible.

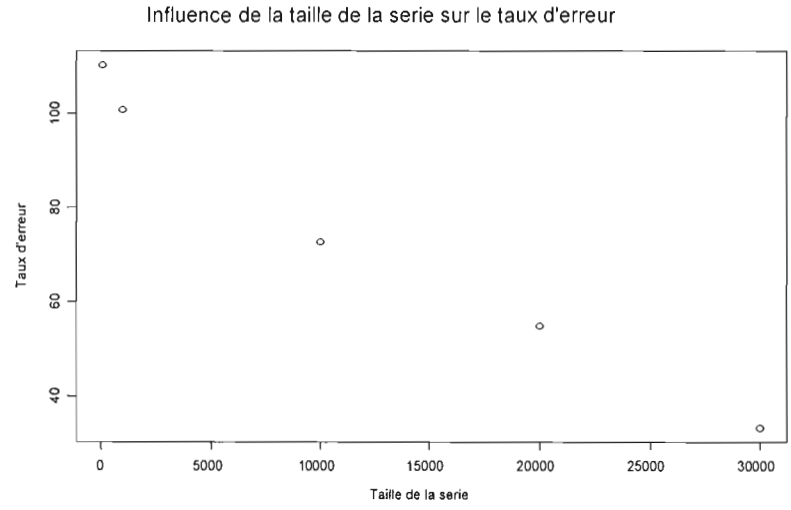


Figure 5.6 Influence de la taille de la série sur le taux d'erreur

5.2.2.4 Simulation de la série avec FGN et des H variées

D'après les simulations faites sur le logiciel statistique *S-PLUS*, sur des séries générées par un processus *FGN*, en variant la valeurs de Hurst H sur des séries de taille allant de *100* à *30000* valeurs, on pourrait conclure que : Pour une série de taille *100* valeurs et/ou *10000* valeurs, tableau *5.9* et tableau *5.11*, les meilleurs taux d'erreurs sont donnés lorsque $H=0.85$, on remarque que la moyenne de taux d'erreurs diminue chaque fois qu'on fait augmenter H jusqu'à *0.85*, mais cette moyenne de taux d'erreur a une tendance à augmenter lorsque H devient supérieure à *0.85*.

Taille de la série égale à 100

Hurst	Le taux d'erreur résultant pour les 10 prochaines valeurs (en %)									
0.50	43.49	110.9	77.06	105.9	88.34	103.0	94.01	101.58	96.92	100.8
0.70	88.19	115.5	99.19	10.73	99.87	100.3	99.97	60.06	99.99	100.0
0.80	122.6	52.89	108.4	62.27	103.1	93.32	31.19	97.48	100.4	99.05
0.85	43.61	64.46	29.93	61.56	27.83	61.13	27.54	61.09	27.54	61.11
0.90	134.1	80.71	100.8	100.2	99.92	100.0	99.53	100.0	100.0	99.99
0.95	195.7	146.7	733.5	135.4	608.3	132.2	570.8	131.2	556.1	130.7

Tableau 5.9 Les 10 prochaines valeurs de l'erreur en variant H

Hurst H	0.50	0.70	0.80	0.85	0.90	0.95
Moyenne_erreur	92.22	87.39	87.07	46.58	101.53	334.10

Tableau 5.10 Le taux moyen d'erreur en variant *H*

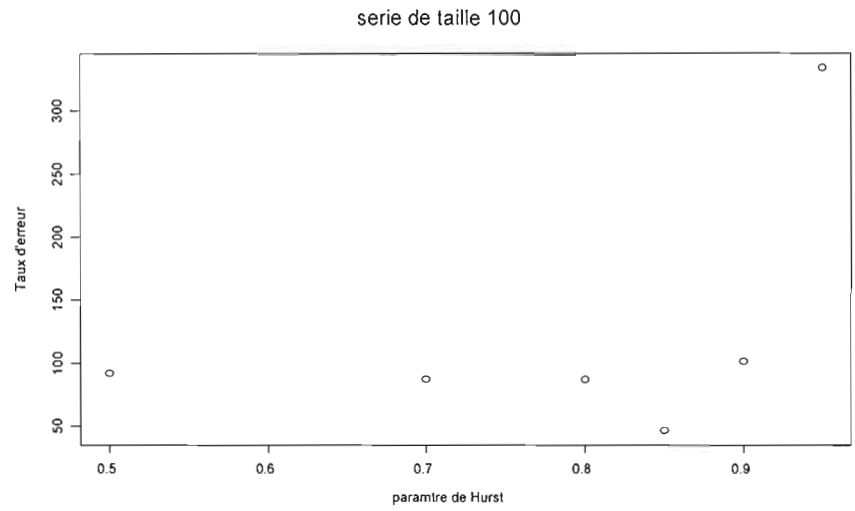


Figure 5.7 Influence de *H* sur le taux d'erreur - série de taille 100

Pour une série de taille *30 000* valeurs, tableau 5.13 et tableau 5.14, les meilleurs taux d'erreurs (la moyenne dans ce cas est égale à *14*) sont données lorsque *H=0.95*, on remarque que la moyenne de taux d'erreurs diminue chaque fois qu'on fait augmenter *H* de *0.5* jusqu'à *0.95* (pour pouvoir conserver les caractéristiques d'un processus auto - similaire), mais cette moyenne de taux d'erreur augmente lorsque *H*

devient supérieure à *0.95*, dans notre cas pour une valeur de H égale à *0.99*, le taux d'erreur est de l'ordre de plus de *350%*.

Taille de la série égale à 10000

Hurst	*Le taux d'erreur résultant pour les 10 prochaines valeurs (en %)*									
0.50	86.45	100.10	99.85	100.00	99.99	100.00	99.99	100.00	99.99	100.00
0.70	99.62	126.4	94.28	127.2	95.01	123.0	95.80	119.3	96.47	16.26
0.85	58.83	29.57	64.42	113.6	65.84	107.7	66.63	103.2	67.32	99.12
0.95	212.1	142.7	144.4	121.5	128.0	116.1	123.4	114.3	121.5	113.44

Tableau 5.11 Les 10 prochaines valeurs de l'erreur en variant H

Hurst H	0.50	0.70	0.85	0.95
Moyenne_erreur	98.63	99.35	77.64	133.77

Tableau 5.12 Le taux d'erreur moyen en variant H

Taille de la série égale à 30000

Hurst	Le taux d'erreur résultant pour les 10 prochaines valeurs (en %)									
0.60	81.56	57.88	90.57	70.93	93.27	79.20	95.18	85.11	96.55	89.34
0.70	72.17	127.6	34.52	28.18	40.36	125.1	47.12	62.23	53.17	119.6
0.85	46.85	16.87	67.61	10.33	22.02	18.32	23.94	23.25	75.43	27.52
0.90	44.51	61.10	46.65	62.34	48.17	23.36	49.54	24.31	30.85	35.24
0.95	20.78	9.28	19.91	9.83	19.26	10.29	18.68	10.71	18.12	11.13
0.99	45.61	47.01	57.53	54.83	61.67	57.78	63.42	59.23	64.44	60.22

Tableau 5.13 Les 10 prochaines valeurs de l'erreur en variant *H*

Hurst H	0.60	0.70	0.85	0.90	0.95	0.99
Moyenne _Erreur	83.95	71.02	33.21	42.60	14.79	57.17

Tableau 5.14 Le taux d'erreur moyen en variant *H*

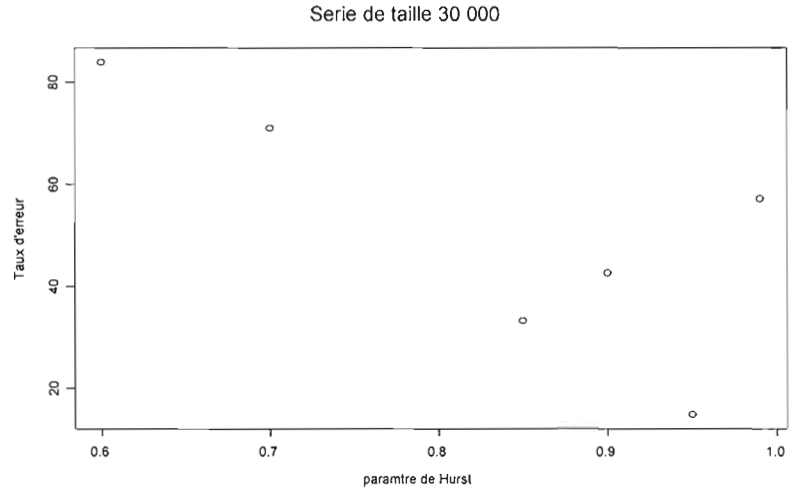


Figure 5.8 Influence de *H* sur le taux d'erreur - série de taille 30,000

Pour conclure, le modèle *ARIMA* ne représente pas dans notre cas, le modèle approprié pour un trafic auto-similaire, parce qu'il donne des résultats qui ne sont pas souvent bons malgré qu'on l'a testé sur trois séries simulées différemment qui représente un trafic auto – similaire.

Le modèle *ARIMA* nous a donné quelques fois les résultats désirés, mais la plupart des temps, les taux d'erreurs ne sont pas les minimums absolus possibles.

5.2.2.5 Quelques travaux récents sur la prédiction avec le modèle *ARIMA*

À partir des données prises d'un trafic *NSFNET* [2], on remarque que les valeurs de cette liste forment un processus non stationnaire, le modèle de série de temps qui pourrait répondre à ce genre de trafic est le modèle *ARIMA*.

Des tests ont été effectué sur des données qui ont été prises entre Août *1988* et Juin *1993*, et pour appliquer le modèle *ARIMA* en prédiction pour une période d'un an avec le modèle *ARIMA*, on doit donner comme liste initiale l'ensemble de données

prises entre Août *1988* et Juin *1992*, puis faire la prédiction de trafic pour un an (de Juillet *1992* jusqu'à Juin *1993*), puis comparer la prédiction durant un an, utilisant le modèle *ARIMA*, avec les valeurs réelles déjà prises à partir du trafic pendant ces *12* mois.

Les figures suivantes, montrent la trajectoire du trafic estimé par rapport au trafic réel pour la période entre Juin *1992* et Juin *1993* avec un taux moyen d'erreur égale à *35 %*.

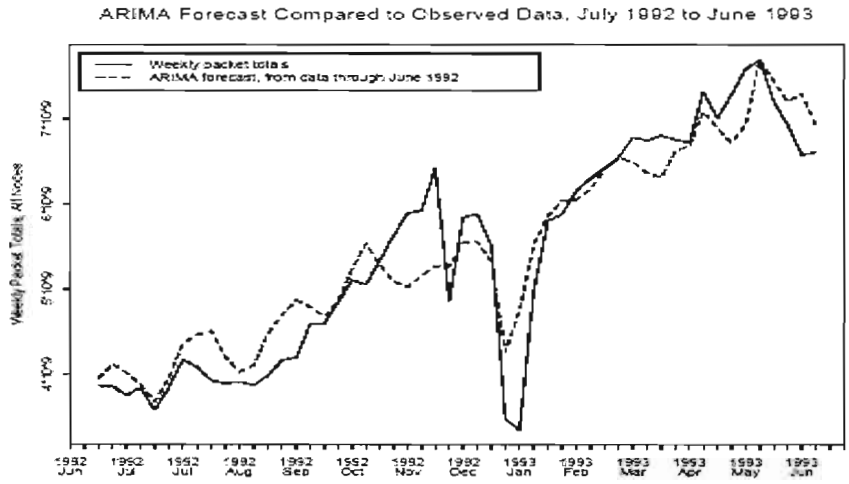


Figure 5: Forecasting with Data from 1988 through June 1992

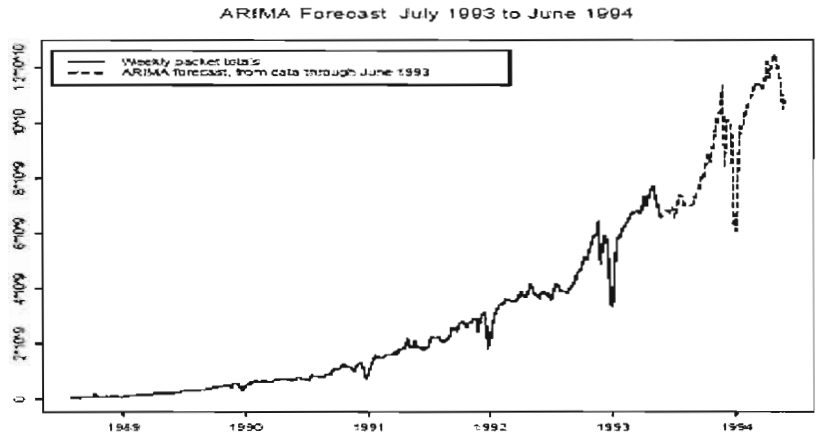


Figure 6: Forecasting with Data from 1988 through June 1993

D'autres travaux ont été fait [1], à partir des observations initiales sur des mesures prises d'un trafic réel d'un réseau Internet, en collectionnant les informations entre deux PoPs, en utilisant *SNMP*, et en se basant sur l'analyse multi- resolutionnelle des ondes et sur les modèles de séries chronologiques linéaires à partir de l'ensemble de mesures prises, on remarque la présence d'une forte périodicité, une tendance à long terme ainsi qu'une variabilité sur des multiples échelles de temps.

Des analyses ont montrés que le signal est capturé par deux composants :

- une tendance à long terme
- et, une fluctuation sur des échelles de temps de taille *12h*

Papagiannaki, Taft, Zhang, Diot [1] ont modélisé l'approximation hebdomadaire des deux composants, en utilisant le modèle *ARIMA*, où ils ont développé un schéma de prédiction qui se base sur leurs comportements prévus.

Construire un modèle de série de temps c'est de trouver l'expression de X_t en fonction des observations précédentes X_{t-j} et de l'événement externe Z_t d'espérance

nulle et de variance finie.

Le modèle approprié qui définie la prédiction du trafic est donnée par :

$$\hat{x}(j) = l(j) + 3dt_3(j)$$

où : - j est l'indice des semaines.

- dt_3 représente la déviation standard hebdomadaire, calculé à partir de la moyenne des sept valeurs d_3 prises chaque semaine. Ce qui donne une valeur affectée à chaque semaine. Avec d_3 le '*detail signal*' à un échelle de temps de *12* heures.

- et *l(j)* représente la tendance à long terme, connaissant la valeur de la sixième approximation du signal (l'analyse de niveau de résolution est faite jusqu'à $2^6 = 96$ heures. On utilise la 6è échelle de temps comme la plus grande échelle de temps qui fourni la meilleure approximation du signal), on calcule sa moyenne pour chaque semaine qui représentera par la suite, *l(j)* où elle capte la tendance à long terme d'une semaine.

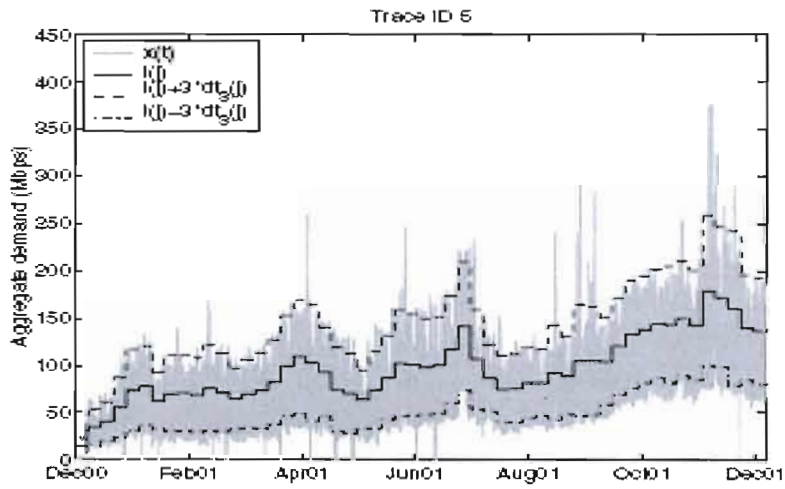


Figure 5.9 Prédiction utilisant le modèle crée entre Dec00 et Dec01

L'objectif suivant est de modéliser les composants : dt_3 et $l(j)$ par le modèle de série chronologique *ARIMA*.

Pour modéliser $l(j)$ et $3dt_3(j)$ en utilisant un modèle de série de temps linéaire, on doit séparer les mesures collectées en : une partie pour estimer les paramètres du modèle, et une autre partie pour évaluer la performance du modèle.

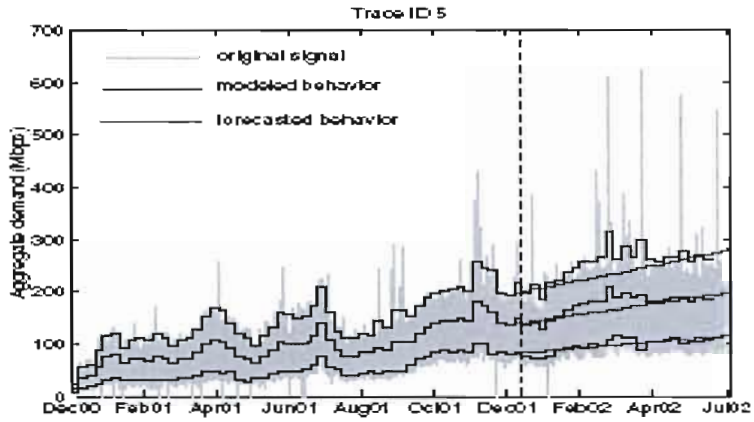


Figure 5.10 Prédiction utilisant le modèle crée entre Dec00 et Jul02

D'après les tests, Figure 5.11, l'erreur de prédiction varie selon le nombre de semaine à prévoir. Par exemple, l'erreur est de l'ordre de *4%* pour une prédiction de *24* semaines (*6* mois), mais elle est de l'ordre de *25%* pour une prédiction de *12* semaines (*3* mois), en moyenne l'erreur de prédiction est de l'ordre de *15%*.

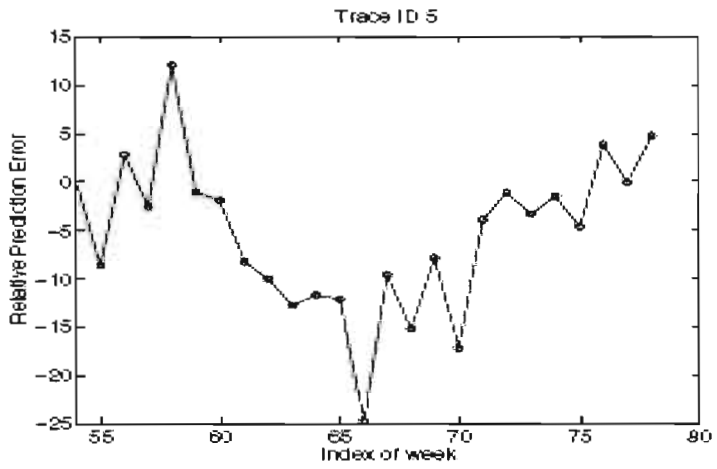


Figure 5.11 Influence de nombre de semaines sur l'erreur de prédiction

5.2.3 Le modèle *F-ARIMA*

La façon la plus classique pour représenter ou modéliser une série qui a un comportement de mémoire longue (*LRD*) est d'utiliser le modèle d'un processus *F-ARIMA* (Fractionally Autoregressive Integrated Moving Average), qui est introduit par Granger et Joyeux (*1980*) et Hosting (*1981*).

On est capable d'utiliser le modèle *F-ARIMA* dans la modélisation d'un trafic auto-similaire qui possède les caractéristiques d'un processus à dépendance à long terme (*LRD*), ainsi que sur un processus à dépendance à court terme (*SRD*), cette propriété a été expérimentée par les auteurs de l'article[16].

5.2.3.1 Définitions

Définition 1 : [16] - Soit X_t est un processus stationnaire, qui vérifie :

$$\Phi(B)(1-B)^d (X_t - \mu) = \Psi(B)\, \varepsilon_t \qquad \text{Eq-5.2.3.1}$$

où d un nombre fractionnaire, et μ est la moyenne du processus,

$$\Phi(B) = 1 - \sum_{j=1}^{p} \Phi_j B^j$$

et $$\Psi(B) = 1 + \sum_{j=1}^{q} \Psi_j B^j$$

sont des polynômes n'ayant pas de racines en dehors du cercle unité et (ε_t) est un bruit blanc centré de variance σ^2

dans ce cas, X_t est appelé un processus *F-ARIMA(p,d,q)*.

Remarquons que si d est un entier positif ou nul, alors l'équation (Eq-5.2.3.1) définit un processus *ARIMA(p,d,q)*.

Le paramètre d dans un processus *F-ARIMA(p,d,q)* est l'indicateur de la force de la dépendance à long terme [16], la relation qui relie ce paramètre à celui de Hurst H est : $H = d + 0.5$.

Définition 2 : [11] - Soit (X_t) un processus *F-ARIMA(p,d,q)* alors :

(i) si $d<1/2$ alors (X_t) est stationnaire

(ii) si $d>-1/2$ alors (X_t) est inversible

(iii) si $0<d<1/2$ alors (X_t) est à mémoire longue

De plus, si $d=0$, le processus (X_t) est à mémoire courte (modèle *ARMA*)

Si $-1/2<d<0$, le processus (X_t) est dit à mémoire intermédiaire.

Si $d>1$, la série originale (X_t) n'est pas stationnaire. Pour obtenir un processus stationnaire, il faut dériver d fois.

En utilisant la fonction gamma, notée $\Gamma(.)$, $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t}\, dt$

On peut simplifier, $(1-B)^d = \sum_{k=0}^{d} C_k^d (-1)^k B^k$

avec $C_k^d = \frac{d!}{k!(d-k)!} = \frac{\Gamma(d+1)}{\Gamma(k+1)\Gamma(d-k+1)}$

Sous la condition d'auto – similarité ($0<d<1/2$), il existe alors une solution stationnaire unique de l'équation : $\Phi(B)(1-B)^d (X_t - \mu) = \Psi(B)\varepsilon_t$

et si le processus est fractionnaire intégré ($\Phi(B) = \Psi(B) = I$), le processus (X_t) pourrait s'écrire sous la forme $X_t = \sum_{k=0}^{\infty} a(k)\varepsilon_{t-k}$

où ε_t sont des variables aléatoires iid, et les coefficients moyenne mobile $a(k)$ sont donnés par :

$$a(k) = \frac{\Gamma(k+d)}{\Gamma(d)\Gamma(k+1)}$$

En particulier pour $k \to \infty$, $a(k) \to \frac{k^{d-1}}{\Gamma(d)}$.

Avec la condition d'inversibilité : $-1/2<d$, le processus (X_t) définie par la définition 1 peut s'écrire sous une forme autorégressive infinie :

$$\sum_{k=0}^{\infty} b_k(d) X_{t-k} = \varepsilon_t$$

où les poids $b_k(d)$, appelés coefficients autorégressifs, sont écrit sous la forme suivante : $b_k(d) \sim \frac{k^{d-1}}{\Gamma(d)}$ *pour* $k \to \infty$

En utilisant la fonction Gamma, le polynôme $(I-B)^d$ se développe sous la forme suivante :

$$(I\text{-}B)^d = \sum_{k\geq 0} b_k(d) B^k$$

où, pour tout k, on a : $$b_k(d) = \frac{\Gamma(k+d)}{\Gamma(d)\Gamma(k+1)}$$

5.2.3.2 Remarques

Pour $0 < d < ½$, qui représente la condition de longue mémoire, on peut remarquer que :

- $\sum_k a_k^2(d) < \infty$, ce qui caractérise la propriété d'un processus stationnaire.

- $\sum_k | a_k(d) | = \infty$, ce qui caractérise la propriété de longue mémoire.

Ainsi par opposition à un processus linéaire [14], par exemple de type *ARMA*, pour lequel par définition $\sum_k | a_k(d) | < \infty$, un processus à mémoire longue *F-ARIMA*, est considéré comme un processus non linéaire.

Deux procédures ont été développées [14], pour estimer les coefficients autorégressifs et les coefficients moyenne mobile en les utilisant dans le logiciel statistique *S-PLUS*, *fracdiff.ar.coef()* et *fracdiff.ma.coef()*. Ces deux procédures prennent comme arguments, le paramètre d, et le nombre de coefficient à utiliser. Ces deux procédures seront utilisées aussi pour réaliser des prédictions sur un trafic auto-similaire.

5.2.3.3 Propositions [11]

Soit (X_t) un processus *F-ARIMA(p,d,q)* stationnaire et inversible, alors sa densité spectrale est donnée par :

$$f_X(\lambda) = \frac{\sigma^2}{2\pi} \frac{|\Psi(e^{i\lambda})|^2}{|\Phi(e^{i\lambda})|^2} |1 - e^{-i\lambda}|^{-2d}$$

Et lorsque $\lambda \rightarrow 0$, $f_X(\lambda) \rightarrow \frac{\sigma^2}{\pi} \frac{|\Psi(1)|^2}{|\Phi(1)|^2} |\lambda|^{-2d}$

Ainsi, si $0<d<1/2$, le processus *F-ARIMA(p,d,q)* est à mémoire longue au sens de la définition dans le domaine spectrale.

La fonction d'autocorrélation d'un processus *F-ARIMA(p,d,q)* est simple à exprimer, lorsque $p=q=0$,

$$\rho(k) = \frac{\Gamma(1-d)\Gamma(k+d)}{\Gamma(d)\Gamma(k+1-d)}$$

Dans le cas général, la fonction d'autocorrélation est trouvée en développant le polynôme $\Phi(B)\Psi^{-1}(B)$ et l'expression asymptotique est donnée par :

$$\rho(k) = C|k|^{2d-1} \ (|k| \rightarrow \infty), \ \text{où } C > 0.$$

Ainsi, si $0<d<1/2$, le processus *F-ARIMA* est à mémoire longue.

5.2.3.4 Simulation d'un processus *F-ARIMA*

En pratique, le logiciel *S-PLUS* permet de simuler des trajectoires finies modélisées par un processus *F-ARIMA(p,d,q)*, à l'aide de la fonction prédéfinie *arima.fracdiff.sim()*, tout en connaissant la moyenne et la variance du processus. Le logiciel S-PLUS fourni cette fonction. Deux autres méthodes ont été proposées utilisant la théorie de transformée rapide de fourrier appliquée sur la fonction d'auto-covariance du processus, et qui représente une grande efficacité en terme de temps d'exécution. Ces deux méthodes sont bien présentées par Beran [17], un programme S qui est fourni et qui permet l'implémentation de ces deux procédures dans *S-PLUS*.

Dans le cas d'un processus *F-ARIMA(0,d,0)* avec $d = 0.3$, $\Phi = 0.2$, $\Psi = 0.1$ et une valeur nulle pour la moyenne du processus, d'après la définition ce processus est à mémoire longue avec H égale à 0.8.

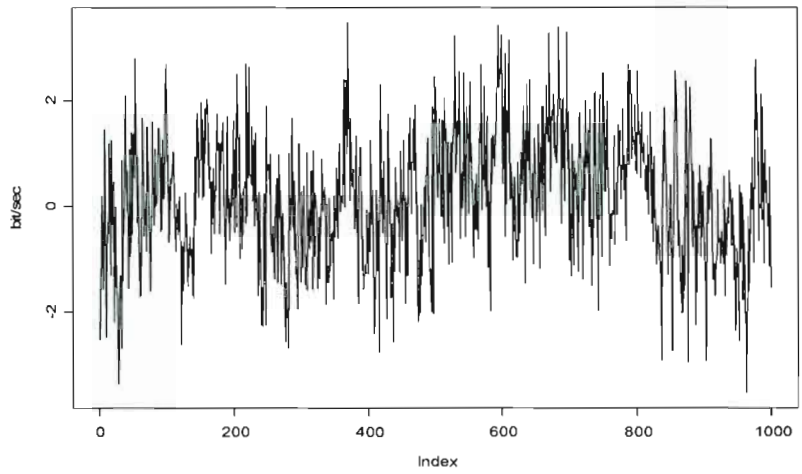


Figure 5.12 Trajectoire d'un processus F-ARIMA

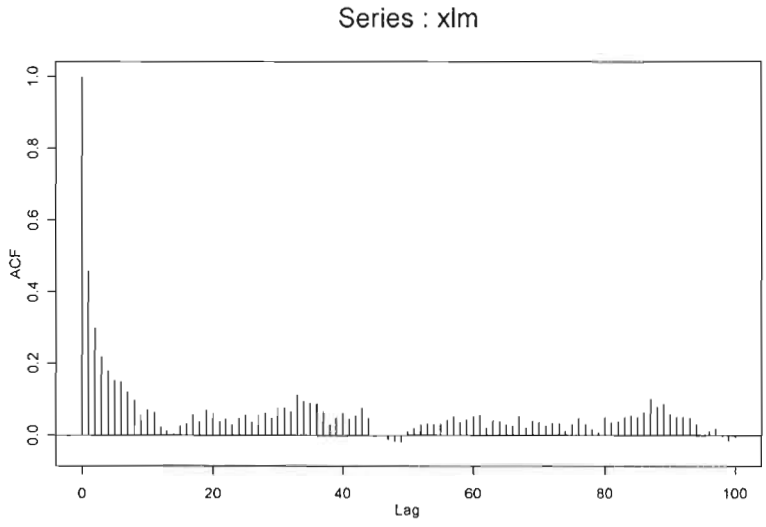


Figure 5.13 Fonction d'autocorrélation d'un processus FARIMA(0,d,0)

On remarque bien qu'à partir de la figure 5.13, la présence d'une lente décroissance de la fonction d'auto- corrélation empirique lorsque la fréquence tend vers

zéro.Il y a aussi la présence d'un cycle de périodicités différentes, bien que la trajectoire du processus soit stationnaire.

5.2.3.5 Prédiction

On s'intéresse maintenant à l'utilisation de processus à mémoire longue *FARIMA* pour effectuer des prévisions sur une série chronologique qui possède les caractéristiques d'un trafic auto-similaire. Une application a été faite dans ce cadre par Collet et Guegan [11] qui ont abordé la prévision des processus à mémoire longue non gaussiens. Ils ont montré l'efficacité des modèles à longue mémoire pour la prévision à moyen et à long terme, mais ils soulignent également les bons résultats obtenus avec les modèles à mémoire courte pour la prévision à court terme.

* Définitions :

À partir des observations X_{n-1}, X_{n-2}, ..., X_1 on calcule le prédicteur $\hat{X}(h)$ à l'horizon h $(h>0)$, ce prédicteur est obtenu en minimisant l'erreur quadratique moyenne de prévision.

Considérons un processus *F-ARIMA(p,d,q)* stationnaire inversible donné par l'équation : $\Phi(B)(1-B)^d (X_t - \mu) = \Psi(B)\, \varepsilon_t$

Cette équation peut s'écrire sous la forme autorégressive infinie suivante :

$$\sum_{j=0}^{\infty} \lambda_j(d) X_{t-j} = \varepsilon_t$$

où $\lambda j(d)$ sont des poids et qui sont calculés à l'aide de développement du polynôme

$$\Phi(B)\, \Psi(B)^{-1} (1-B)^d \ .$$

En particulier, pour $p=q=0$, on aura :

$$\lambda j(d) = b_j(d) = \frac{\Gamma(k-d)}{\Gamma(-d)\Gamma(k+1)}$$

Alors, le prédicteur optimale au temps t pour l'horizon h est donnée par :

$$\hat{X}_t(h) = -\sum_{j=0}^{h-1} \lambda_j(d)\hat{X}_t(h-j) - \sum_{j>h-1}^{\infty} \lambda_j(d) X_{t+h-j}$$

Mais malheureusement, contrairement à *ARIMA*, le logiciel *S-PLUS* ne possède pas une fonction prédéfinie qui permet d'effectuer des prévisions avec un processus *F-ARIMA*. Cependant, en pratique, un algorithme de prévision est implémenté par une méthode qui prend comme paramètres obligatoires le nom de la série initiale à utiliser pour la prédiction, la valeur du paramètre de mémoire d ainsi que de l'horizon de prévision h.

Pour prévoir les futures observations par rapport à un ensemble fini de valeurs prises du passé, on utilise la formule de 'Best Linear Predictor' (Hosking 1981).

* Propositions :

Soit X_t un processus *F-ARIMA* (0, d, 0) avec $-1/2<d<1/2$, dans notre cas, on prendrait d tel que : $0<d<1/2$.

L'équation $\hat{X}_t = \sum_{j=1}^{k} \beta_{kj} X_{t-j}$ représente le meilleur prédicteur linéaire de X_t tout en connaissant X_{t-1} , ..., X_{t-k} , avec β_{kj} sont les coefficients (qui minimisent l'erreur de prédiction $E\left[(X_t - \hat{X}_t)^2 \mid X_{t-1},...,X_{t-k}\right]$). Ces coefficients seront exprimés de la manière suivante : $\beta_{kj} = -C_j^k \dfrac{\Gamma(j-d)\Gamma(k-d-j+1)}{\Gamma(-d)\Gamma(k-d+1)}$

En particulier, pour j, $k \to \infty$ et $j/k \to 0$, on aura :

$$\beta_{kj} \to \frac{j^{-d-1}}{\Gamma(-d)}$$

5.2.3.6 Prédiction avec *FARIMA* sur des séries simulées par *F-ARIMA (p, d, q)*

Les tests ont été faits pour une valeur de paramètre de Hurst *H* égale à *0.85*, qui caractérise bien un trafic auto similaire, et appliquée sur des séries chronologiques simulées par un processus à mémoire longue *F-ARIMA* de taille initiale variable allant de *100* jusqu'à *30000* afin de prévoir les *10* prochaines valeurs. Le tableau suivant nous montre les résultats obtenus pour les *10* futurs taux d'erreurs

Taille	Le taux d'erreur résultant pour les 10 prochaines valeurs (en %)									
100	85.05	79.84	81.54	77.60	82.69	140.5	107.7	146.4	187.7	213.1
1000	141.0	11.90	139.6	185.8	132.8	18.22	179.7	220.3	143.6	10.05
10000	44.81	7.55	2.67	20.41	69.18	65.34	40.36	36.12	31.18	63.33
20000	19.80	14.10	16.21	46.48	10.88	31.93	35.97	38.81	86.52	74.40
30000	18.40	58.95	9.77	17.01	31.54	25.70	7.75	18.34	14.26	61.58

Tableau 5.15 Les 10 prochaines valeurs de l'erreurs en variant la taille - FARIMA

Taille_serie	100	1000	10000	20000	30000
Moyenne_erreur	120.24	118.33	38.09	37.51	26.33

Tableau 5.16 Le taux moyen d'erreur en variant la taille - FARIMA

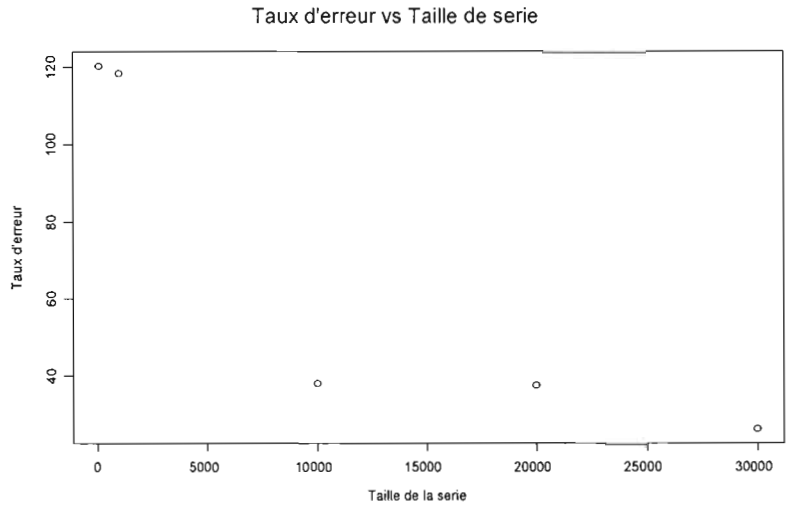


Figure 5.14 Influence de la taille de la série sur le taux d'erreur

On remarque bien que la moyenne de taux d'erreur diminue chaque fois qu'on fait augmenter la taille de la série simulée. Ce qui nous permet de conclure qu'on pourrait avoir une meilleure prédiction désirée si on se base sur un historique de taille assez grande. D'après le Tableau 5.16, le modèle réalise une prédiction assez 'bonne' de l'ordre de *26.33%* dans le meilleur des cas. Pour ce, on s'intéresserait par la suite aux séries chronologiques de taille supérieure à *30 000* valeurs en variant à chaque fois la valeur du paramètre de Hurst H afin de trouver la meilleur valeur qui pourrait être affecter à *H* pour avoir un meilleur taux d'erreur utilisant le modèle *F-ARIMA*.

La taille de la série simulée est de l'ordre de *30 000*, en se basant sur cette liste initiale, on fait varier la valeur du paramètre de Hurst *H*, si dessous les résultats trouvés pour le taux d'erreur $\frac{|\hat{X}_{i+1} - X_{i+1}|}{|X_{i+1}|}$ et pour la moyenne des taux d'erreur :

Taille de la série est de l'ordre de 30 000 valeurs

Hurst	Le taux d'erreur résultant pour les 10 prochaines valeurs (en %)									
0.55	93.38	91.53	101.6	98.81	100.5	99.41	101.0	99.54	100.2	101.7
0.60	130.2	74.53	102.8	22.85	98.30	94.21	101.2	99.37	100.7	97.94
0.70	98.03	10.99	115.7	85.84	101.1	55.02	98.69	16.48	104.4	98.21
0.80	10.05	92.77	15.93	18.38	42.08	103.9	63.70	69.86	15.61	65.70
0.85	18.40	58.95	9.77	17.01	31.54	25.70	7.75	18.34	14.26	61.58
0.90	94.23	76.61	126.0	118.3	11.04	85.82	187.9	29.86	128.4	68.39
0.99	128.9	66.13	186.6	59.04	19.63	48.90	169.9	134.30	320.4	118.1

Tableau 5.17 Les 10 prochaines valeurs de l'erreurs en variant H - FARIMA

Hurst H	0.55	0.60	0.70	0.80	0.85	0.90	0.95
Moyenne_erreur	98.79	92.22	78.45	49.80	26.33	92.67	125.21

Tableau 5.18 Le taux moyen d'erreur en variant H - FARIMA

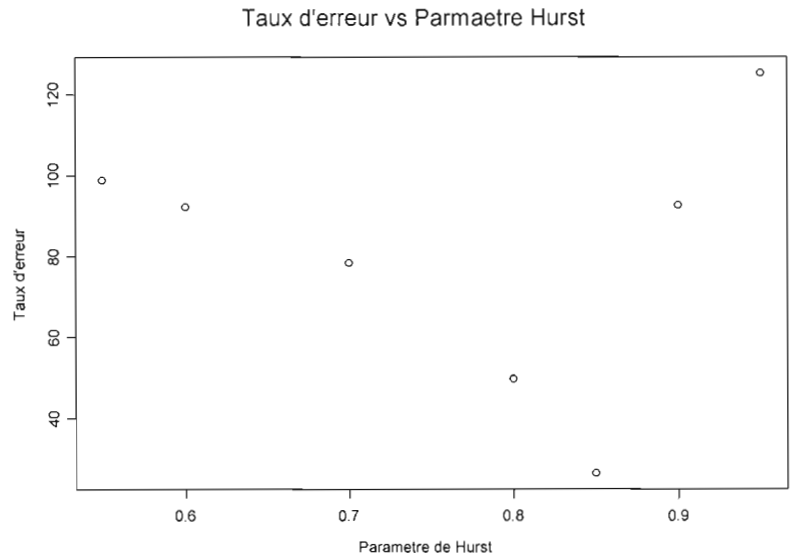


Figure 5.15 Le taux moyen d'erreur VS Le paramètre de Hurst

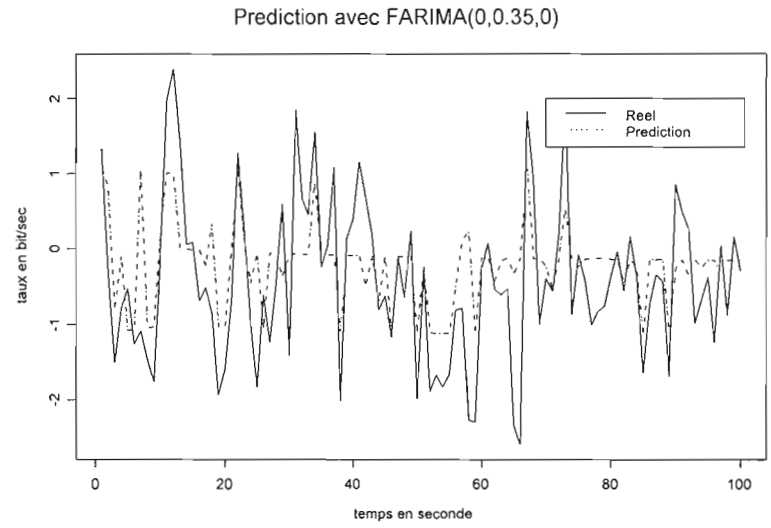


Figure 5.16 Trafic Estimé VS Trafic Réel

Le modèle *F-ARIMA* a montré une grande capacité de fournir les propriétés actuelles du trafic, en lui affectant la valeur *0.85* au paramètre de Hurst *H*. Ce modèle, bien évidemment, pourrait être utilisé pour décrire la dépendance à long terme du trafic Internet. En appliquant ce modèle pour des fins de prédiction de trafic, ce modèle réalise des résultats acceptables, Figure 5.16, ces résultats sont meilleures que celles obtenus par le modèle *ARIMA*.

5.2.4 Le modèle Fractional Brownian Motion (*FBM*)

5.2.4.1 Définition

On considère un processus auto - similaire Y_t à incrémentation stationnaire.

Soit $X_i = Y_i - Y_{i-1}$, X_i est la valeur prévue du processus incrémenté.

La covariance de X_i est donnée par :

$$\gamma(k) = \frac{1}{2}\ \sigma^2[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}] \quad \text{Eq-5.2.4.1}$$

En particulier, supposons que X_t est un processus Gaussien, alors, la distribution du processus est caractérisée par son espérance et sa covariance.

Pour chaque valeur de H tel que $0 < H < 1$, il existe un et un seul processus Gaussien X_i. Ce processus est connu sous le nom de Fractional Gaussian Noise (*FGN*). Le processus auto - similaire Y_i correspondant est connu sous le nom Fractional Brownian Motion, et est noté par $B_H(t)$.

Dans le cas où $H = ½$, les variables X_1, X_2, ... sont des variables normales indépendantes. Le processus auto - similaire correspondant, $B_{1/2}(t)$, est un processus Brownian Motion, noté $B(t)$.

5.2.4.2 Simulation d'un processus *FGN* (Fractional Gaussian Noise)

Afin de simuler un processus ou une série de temps stationnaire Gaussien de taille n et d'auto-covariance $\gamma(0), \gamma(1), \gamma(2) \ldots \gamma(n-1)$, on utilise une méthode qui se base sur la transformé rapide de Fourier (Fast Fourier Transform fft) car cette méthode permet un calcul et une implémentation plus facile et plus rapide.

On définit tout d'abord $\lambda_k = \dfrac{2\pi(k-1)}{2n-2}$ pour $k=1, \ldots, 2n-2$

Puis on estime la valeur de g_k la transformé de fourier de $\gamma(0), \gamma(1), \ldots, \gamma(n-2), \gamma(n-1), \gamma(n-2), \ldots, \gamma(1)$

$$g_k = \sum_{j=1}^{n-1} \gamma(j-1) e^{i(j-1)\lambda_k} + \sum_{j=n}^{2n-2} \gamma(2n-j-1) e^{i(j-1)\lambda_k}$$

pour k allant de 1 jusqu'à $2n-2$.

Puis il faut vérifier que $g_k > 0$ pour tout k allant de 1 jusqu'à $2n-2$.

Ensuite, il faudrait simuler deux séries indépendantes de variables aléatoires normales de moyenne zéro, notées $U_1, U_2, \ldots, U_n$ et $V_1, V_2, \ldots, V_{n-1}$ tels que :

$$Var(U_1) = Var(U_n) = 2$$

et, pour $k \neq (1,n)$

$$Var(U_k) = Var(V_k) = 1$$

avec $V_1 = V_n = 0$ et Z_k des variables aléatoires complexes définies par :

$$Z_k = U_k + iV_k \quad \text{où } k = 1, \ldots, n$$

$$\text{et } Z_k = U_{2n-k} - iV_{2n-k} \quad \text{où } k = n+1, \ldots, 2n-2$$

Enfin, en définit la série résultante X_t qui est une série Gaussienne stationnaire

$$X_t = \frac{1}{2\sqrt{n-1}} \sum_{k=1}^{2n-2} \sqrt{g_k}\, e^{i(t-1)\lambda_k} Z_k$$

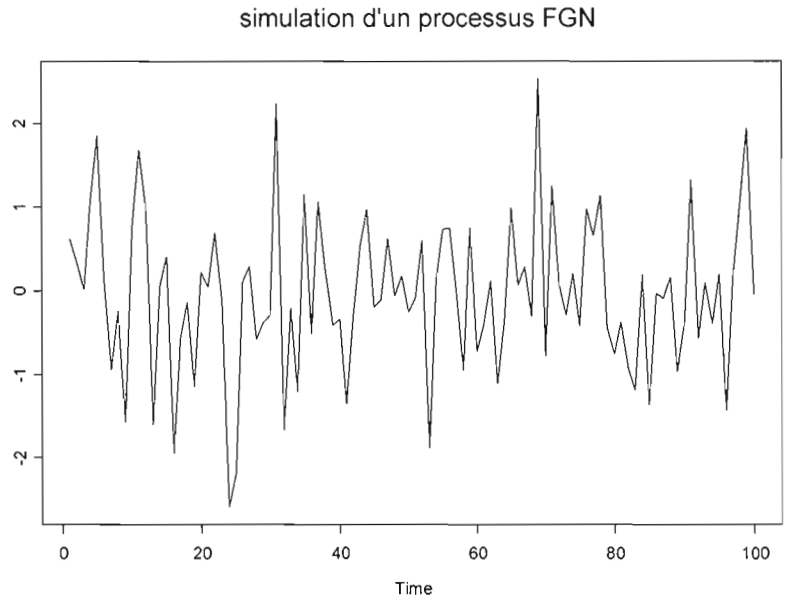


Figure 5.17 Simulation d'un processus FGN

5.2.4.3 Propositions

* Soit *B(t)* un processus stochastique qui vérifie :

(i) *B(t)* est Gaussien

(ii) $B(0) = 0$

(iii) $E[B(t) - B(s)] = 0$

(iv) $var[B(t) - B(s)] = \sigma^2 |t\text{-}s|$

(v) *B(t)* possède une incrémentation indépendante.

alors *B(t)* est appelé un processus Brownien.

De (ii) et (iii) on a : $E[B(ct)] = E[B(ct)] - B(0)] = 0 = c^{\frac{1}{2}} E[B(t)]$

et $cov(B(t), B(s)) = var(B(s)-B(0)) = 0 = \sigma^2 s = \sigma^2 min(t,s)$

puisque $[B(t)-B(s)]$ est indépendant de $B(s)-B(0)=B(s)$

et $cov(B(ct), B(cs)) = c\sigma^2 min(t,s) = cov(c^{1/2}B(t), c^{1/2}B(s))$

alors, $B(t)$ est un processus auto -similaire avec un paramètre d'auto -similarité

$$H = ½.$$

Un processus *FBM* (Fractional Brownian Motion) avec le paramètre H est définie par la fonction de covariance : l'équation (Eq-5.2.4.1).

D'autres hypothèses ont été ajoutées par Adrian [13] à la définition proposée par Beran :

(i) $B_H(t+\delta) - B_H(t)$ est $N(0, \sigma \mid \delta \mid^H)$

(ii) $E(B_H(t)B_H(s)) = \sigma^2 / 2(\mid t \mid^{2H} + \mid s \mid^{2H} - \mid t-s \mid^{2H})$

l'hypothèse (ii) montre que $Var(B_H(t)) = \sigma^2 \mid t \mid^{2H}$

$B_H(t)$ est exactement auto –similaire de paramètre H.

Pour créer une simulation d'un processus *FBM*, on pourrait utiliser un script Matlab.

* Une définition mathématique pourrait exprimer la définition mais en terme d'intégrale :

Soit $s > 0$, on définit la fonction ω_H par :

$$\omega_H (t, u) = 0 \quad \text{pour } t \leq u$$

$$\omega_H\ (t\ ,\ u\ ) = (\ t - u\ )^{H-1/2} \quad \text{pour} \quad 0 \le u < t$$

$$\text{et} \quad \omega_H (\ t\ ,\ u\ ) = (\ t - u\ )^{H-1/2} - (-u)^{H-1/2} \quad \text{pour } u < 0$$

aussi, soit $B(t)$ un processus *FBM* Standard (définition 1), avec $\sigma^2 = 1$.

pour $0 < H < 1$, $B_H(t) = s\int \omega_H(t,u)dB(u)$

d'où $B_H(t)$ est appelé un processus *FBM* (Fractional Brownian Motion) de paramètre d'auto – similarité H.

* Un modèle *FBM* normalisé avec un paramètre de Hurst $H \in [1/2, 1]$ est une série chronologique Z_t vérifiant :

(i) Z_t possède une incrémentation stationnaire

(ii) $Z_0 = 0$, et $E[Z_t] = 0$ pour tout t

(iii) $E[\ Z_t^2\ ] = |\ t\ |^{2H}$ pour tout t

(iv) Z_t est un processus Gaussien

Sa covariance d'incrémentations pour les deux intervalles $[t_1, t_2]$ et $[t_3, t_4]$

(Avec $t_1 < t_2 \le t_3 < t_4$) est toujours positive, et est donnée par :

$$cov(Z_{t_2} - Z_{t_1}, Z_{t_4} - Z_{t_3}) = \frac{1}{2}((\ t_4 - t_1)^{2H} - (\ t_3 - t_1)^{2H} + (\ t_3 - t_2)^{2H} - (\ t_4 - t_2)^{2H}\ )$$

5.2.4.4 Prédiction

La prédiction de Z_a (a>0) basée sur $\{Z_t \mid \ t \in (-T, 0)\}$ est équivalente à la prédiction de la différence $Z_{t+a} - Z_t$ (pour tout t) basée sur la différence $Z_t - Z_s$

avec $s \in (t\text{-}T,t)$, ce qui donne :

le prédicteur $\hat{Z}_{a,T} = E(Z_a \mid Z_s, s \in (-T,0))$ qui pourrait être exprimé comme :

$$\hat{Z}_{a,T} = \int_{-T}^{0} g_T(a,t)dZ_t$$

où la fonction $g_T(a,t)$ est exprimée par :

$$g_T(a,t) = \frac{\sin(\pi(H-\frac{1}{2}))}{\pi}(-t(T+t))^{-H+\frac{1}{2}} \int_0^a \frac{(\tau(\tau+T))^{H-\frac{1}{2}}}{\tau - t} d\tau$$

5.2.5 Le modèle Linear Minimum Mean Square Error *LMMSE*

5.2.5.1 Définitions

Etant donnée une série chronologique f (k), avec $k = 1, \ldots, n$; qui représente la quantité de trafic mesurée durant un intervalle de temps.

Les valeurs de $f^{(m)}(1)$, $f^{(m)}(2)$, ..., $f^{(m)}(n)$ sont mesurées à partir des n intervalles de mesure (de taille τ), où $f^{(m)}(k)$, $1 \leq k \leq n$, représente la série de valeurs agrégées durant le $(n + 1 + k)$ ième intervalle de temps, et pourrait être exprimée comme la moyenne des sommes des m dernières valeurs de la série chronologique.

$$f^{(m)}(k) = \sum_{i=(k-1)m+1}^{km} f(i) \qquad \text{Eq-5.2.5.1}$$

où m est le nombre de valeurs dans un seul intervalle de mesure.

L'estimé de $f^{(m)}(n+1)$ qui est noté $\hat{f}^{(m)}(n+1)$ est donné par :

$$\hat{f}^{(m)}(n+1) = [a_1, a_2, a_3, \ldots a_n] \begin{bmatrix} f^{(m)}(1) \\ f^{(m)}(2) \\ \ldots \\ f^{(m)}(n) \end{bmatrix} \qquad \text{Eq-5.2.5.2}$$

où $a_1, a_2, a_3, \ldots a_n$ sont appelés les coefficient du prédicteur *LMMSE*, ils sont calculés à partir des fonctions d'auto - corrélation.

R (n) représente la fonction d'auto - corrélation de la série chronologique, est estimée dans la pratique (due à propriété d'auto –similarité asymptotiquement de second ordre) par :

$$R(i) \cong R^{(m)}(i) = \frac{1}{n}\sum_{t=i+1}^{n} f^{(m)}(t) f^{(m)}(t-i) \qquad \text{Eq-5.2.5.3}$$

où $0 \leq i \leq n\text{-}1$, et n le nombre de séries de valeurs agrégées. (Dans les simulations et leurs études empiriques, les auteurs de l'article [13] supposent des valeurs de n supérieures à *20*).

Notons que la fonction d'autocorrélation pourrait être calculée facilement [18]-[19] en utilisant l'estimation de séries de valeurs agrégées prises en ligne et du paramètre de Hurst *H*.

Les coefficients du modèle *LMMSE*, sont alors calculés de la manière suivante :

$$[a1, a_2, a_3, \ldots a_n] = [R(n)\ R(n\text{-}1)\ \ldots\ R(1)]\ x \begin{bmatrix} R(0)R(1)\ldots\ldots R(n-1) \\ R(1)R(0)\ldots\ldots R(n-2) \\ \ldots\ldots\ldots\ldots\ldots \\ R(n-1)R(n-2)\ldots\ldots R(0) \end{bmatrix}^{-1} \qquad \text{Eq-5.2.5.4}$$

L'erreur moyenne du prédicteur *LMMSE* est donnée (après quelques opérations algébriques) par :

$$\sigma^2 = \sigma_x^2 - [R(n)R(n-1)\ldots\ldots R(1)]x \begin{bmatrix} R(0)\ldots\ldots\ldots\ldots R(n-1) \\ R(1)\ldots\ldots\ldots\ldots R(n-2) \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ R(n-1)\ldots\ldots\ldots\ldots R(0) \end{bmatrix}^{-1} x \begin{bmatrix} R(n) \\ R(n-1) \\ \ldots\ldots \\ R(0) \end{bmatrix} \qquad \text{Eq-5.2.5.5}$$

σ^2 n'est pas facile à calculer, on pourrait utiliser alors : $R(t) = H(2H-1)t^{2H-2}$.

5.2.5.2 Algorithme à suivre

a- Prendre en entrée un vecteur de n valeurs (qui représente la série chronologique)

b- Calculer les n fonctions de covariance de la série, estimés en pratique par :

$$R(i) \cong R^{(m)}(i) = \frac{1}{n}\sum_{t=i+1}^{n} f^{(m)}(t) f^{(m)}(t-i)$$

c- Calculer les coefficients de *LMMSE*,

$$[a_1, a_2, a_3, \ldots a_n] = [R(n)\ R(n-1)\ \ldots\ R(1)] \times \begin{bmatrix} R(0)R(1).....R(n-1) \\ R(1)R(0).....R(n-2) \\ ..... \\ R(n-1)R(n-2).....R(0) \end{bmatrix}^{-1}$$

d- Obtenir en sortie la (n *ième* $+1$) valeur à prévoir :

$$\hat{f}^{(m)}(n+1) = [a_1, a_2, a_3, \ldots a_n] \begin{bmatrix} f^{(m)}(1) \\ f^{(m)}(2) \\ \ldots \\ f^{(m)}(n) \end{bmatrix}$$

e- Calculer l'erreur de la prédiction: $err = \left| \frac{\hat{f}(n+1) - f(n+1)}{f(n+1)} \right|$

f- Puis trouver de la même façon (*n ième +2*) valeur à prévoir :

$$\hat{f}^{(m)}(n+2) = [a_1, a_2, a_3, \ldots a_n] \begin{bmatrix} f^{(m)}(2) \\ f^{(m)}(3) \\ \ldots \\ f^{(m)}(n+1) \end{bmatrix}$$

5.2.5.3 Implémentation dans Matlab

Afin d'implémenter le prédicteur *LMMSE*, on devrait prendre en considération que :

* Le prédicteur *LMMSE* dérive d'un processus stochastique stationnaire d'espérance zéro, et comme les séries chronologiques sont mesurées on-line et peuvent ne pas être d'espérance nulle, il faudrait commencer par soustraire la moyenne de l'ensemble de valeurs de la série originale, la valeur moyenne de cette série:

(série de temps original) – (valeur moyenne), et par la suite appliquer le prédicteur *LMMSE* pour estimer la série de temps agrégée dans le prochain intervalle puis rajouter la valeur moyenne.

* On est obligé de déterminer à quel point la prédiction du trafic est réalisée, ce qui amène au problème de déterminer la valeur appropriée, τ, qui représente l'intervalle de temps de mesure. Grâce aux caractéristiques de la mémoire longue (*LRD*) du trafic Internet qui donne une lente décroissance de la fonction d'auto – corrélation, on pourrait choisir une valeur assez grande pour τ. Dans la simulation on prend l'intervalle de mesure τ, de l'ordre de $0.1s$ à quelques RTT (Round Trip Time, qui représente le temps nécessaire pour faire un aller retour). Dans notre cas, on a utilisé des intervalles de mesures de taille $0.1s$.

* Appliquant la procédure Hurst sur nos deux séries chronologiques expérimentées « *in_trace* » et « *in_alpha* » dans le but de trouver la valeur affectée au paramètre de Hurst *H*, on a obtenu la valeur *0.749* pour la première série chronologique et *0.754* pour la seconde série chronologique.

* l'opération utilisée pour calculer les coefficients du prédicteur *LMMSE* (Eq-5.2.5.3 et Eq-5.2.5.4) est la multiplication des échantillons de séries chronologiques avec la matrice de fonction d'autocorrélation, cette opération semble être lourde. Mais, on pourrait utiliser comme dans les travaux de [14] qui a utilisé un algorithme plus rapide basé sur la prédiction linéaire où une des équations utilisées permet d'éviter l'opération de multiplication par une matrice. L'algorithme commence d'abord par estimer le premier coefficient a_0. Puis à chaque instant de mesure, un nouveau $f^m(n)$ est obtenu, et l'algorithme calcule récursivement les coefficients $a_1, a_2, a_3, \ldots a_{n+1}$ du prédicteur *LMMSE* utilisant l'équation :

$$a_{n+1} = a_n + \mu.e\,(n).\, f^m(n)\,. \qquad \text{Eq-5.2.5.6}$$

où $e(n)$ est l'erreur de prédiction et μ est une constante

Si f est stationnaire alors a_i converge en moyenne vers la solution optimale [14].

5.2.5.4 Validation du modèle

Pour Valider le modèle conçu qui illustre la prédiction avec la méthode de *LMMSE*, on l'a implémenté dans le logiciel statistique Matlab puis tester sa capacité de prédiction sur deux séries chronologiques « *in_trace* » et « *in_alpha* » qui ont les caractéristiques d'un processus auto-similaire avec des paramètres de Hurst différents. Chaque série contient deux milles valeurs, et chaque série représente des valeurs collectées à partir du trafic Internet réel. Et comme on l'avait précisé précédemment, on devrait travailler sur des séries de valeurs agrégées (Eq-5.25.1), on a choisi dans les deux cas, une valeur de *m* égale à *14* afin d'obtenir *142* intervalles où chaque intervalle contient *14* valeurs de la série chronologique réelle. Les quatre figures suivantes

illustrent la trajectoire de la série originale et de la série de valeurs agrégées pour les deux séries chronologiques « *in_trace* » et « *in_alpha* » respectives.

En ce qui concerne la série chronologique « in_trace », pour une valeur de n de l'ordre de *10*, l'erreur moyenne est égale à *0.0380* et la déviation standard est égale à *0.0313*. Pour la série chronologique « in_alpha », pour une valeur de n de l'ordre de *10*, l'erreur moyenne est égale à *0.1046* et la déviation standard est de l'ordre de *0.0805*.

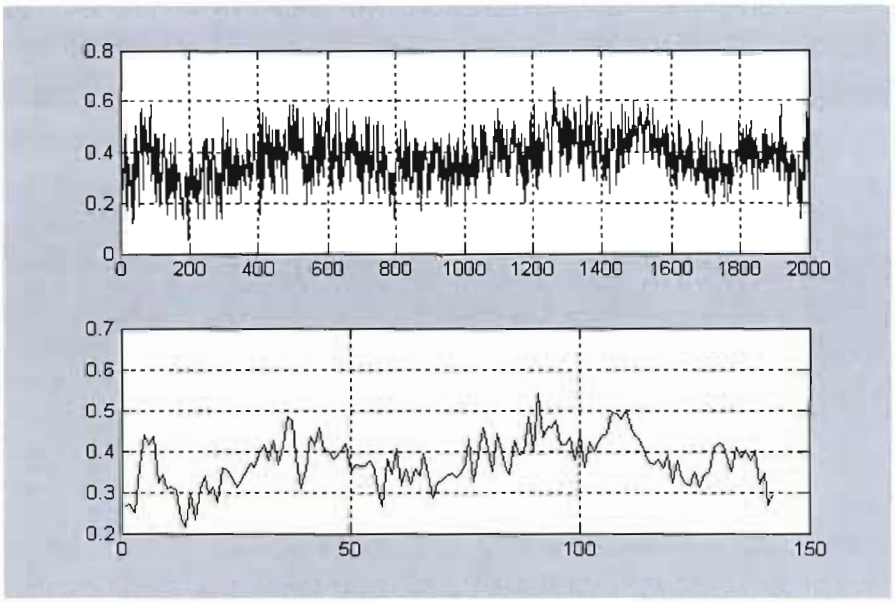


Figure 5.18 La série originale et la série agrégée : in_trace, n=10

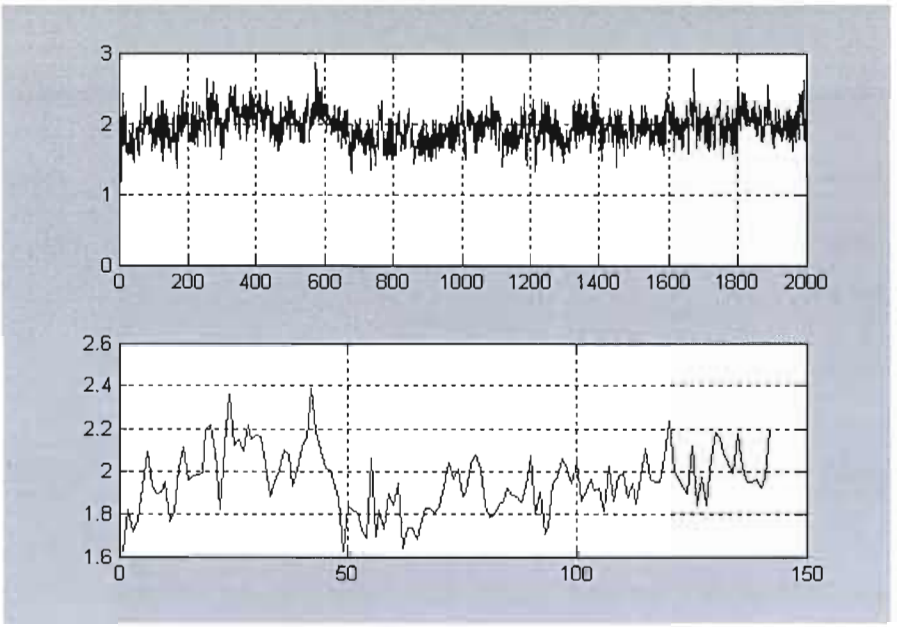


Figure 5.19 La série originale et la série agrégée : in_alpha, n=10

On a obtenu des résultats excellents vue que le trafic réel et le trafic estimé se ressemblent énormément. Pour illustrer cette conclusion, la Figure 5.20 pour la série « *in_trace* » et la Figure 5.21 pour la série « *in_alpha* » montrent bien que le trafic estimé par le modèle *LMMSE* ressemble très bien à celui du trafic réel.

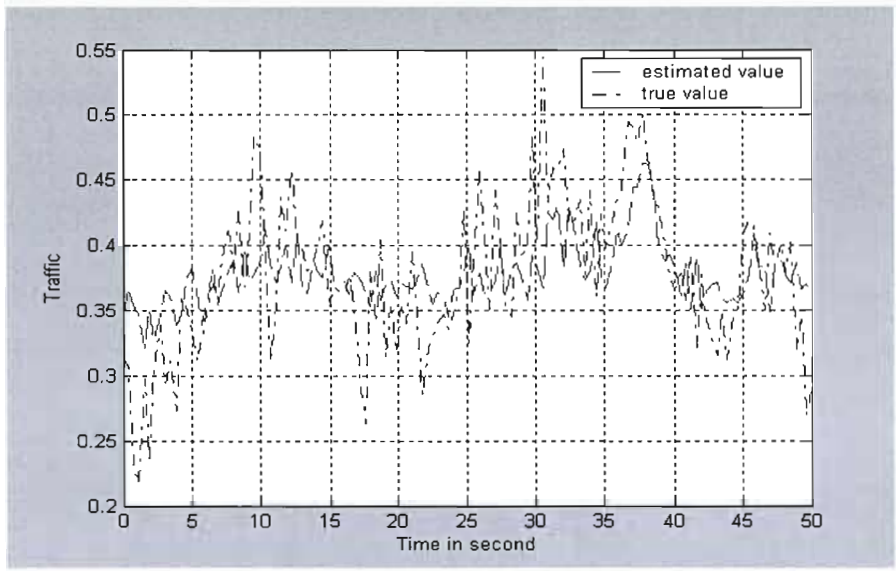


Figure 5.20 Trafic estimé VS Trafic réel avec LMMSE : in_trace, n=10

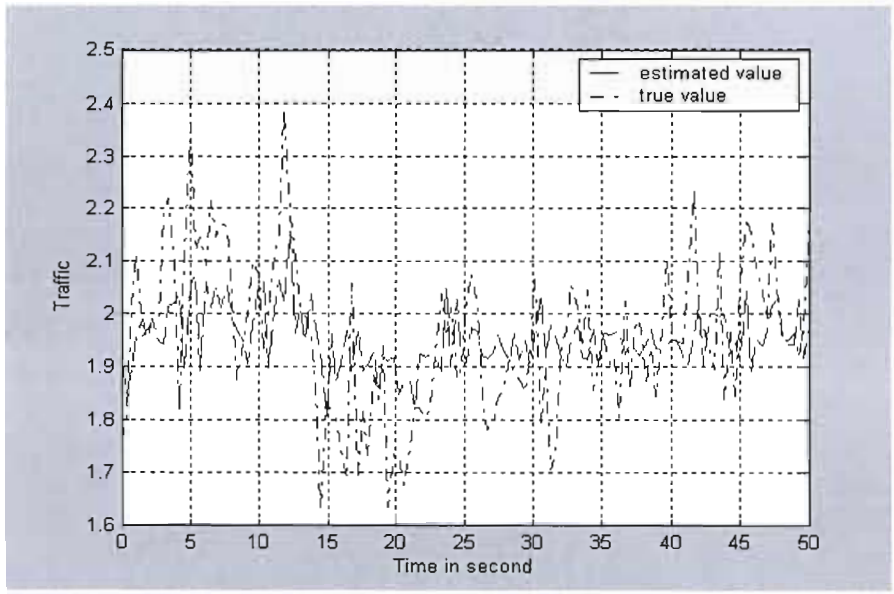


Figure 5.21 Trafic estimé VS Trafic réel avec LMMSE : in_alpha, n=10

Le paramètre de Hurst H dans les deux cas prend les valeurs 0.749 et 0.754 respectivement, ce qui confirme les caractéristiques du trafic auto - similaire et aussi de sa propriété de mémoire à longue portée (LRD).

Le taux d'erreur, donnée par $\left|\frac{\hat{f}(n+1) - f(n+1)}{f(n+1)}\right|$, est de l'ordre de 0.08 pour le premier cas, et de l'ordre de 0.14 pour la deuxième série chronologique.

Dans notre simulation, on a fait changer les valeurs de n (le nombre des valeurs de séries agrégées) de $n = 10$ à $n = 40$, pour une meilleure performance dans la prédiction, on a choisit $n = 20$, (ce qui montre le fait que la fonction d'auto-corrélation décroît lentement), avec cette valeur de n, les résultats obtenus nous permettent d'avoir un meilleur taux d'erreur, ainsi qu'une déviation standard (écart type) inférieure;

En ce qui concerne la série chronologique « in_trace », pour une valeur de n de l'ordre de 20, l'erreur moyenne est égale à 3.42% et la déviation standard est égale à 0.0283. Pour la série chronologique « in_alpha », pour une valeur de $n = 20$, l'erreur moyenne est égale à 0.1006 et la déviation standard est de l'ordre de 0.0803.

Les figures suivantes nous montrent la prédiction pour les deux séries testées, ainsi que la trajectoire du trafic réel versus le trafic estimé lorsque n égale à 20 :

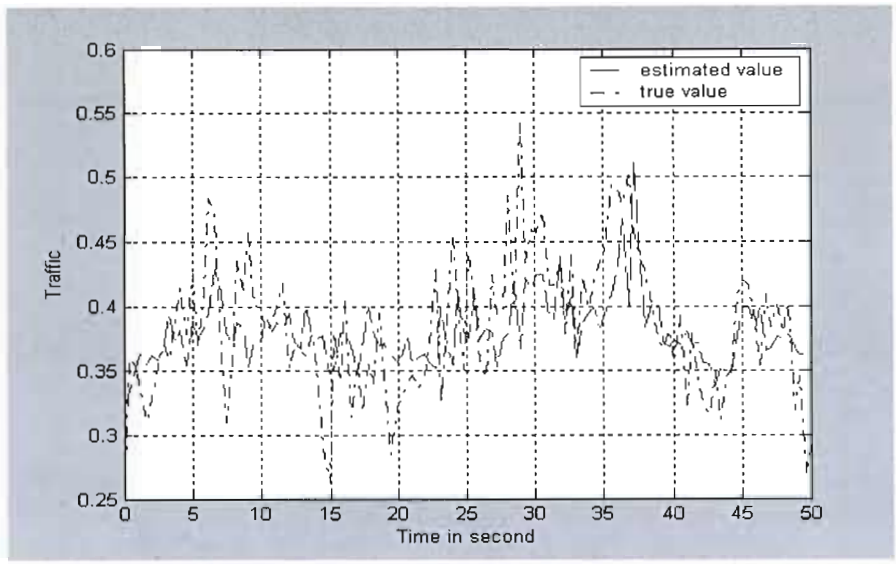


Figure 5.22 Trafic estimé VS Trafic réel avec LMMSE : in_alpha, n=20

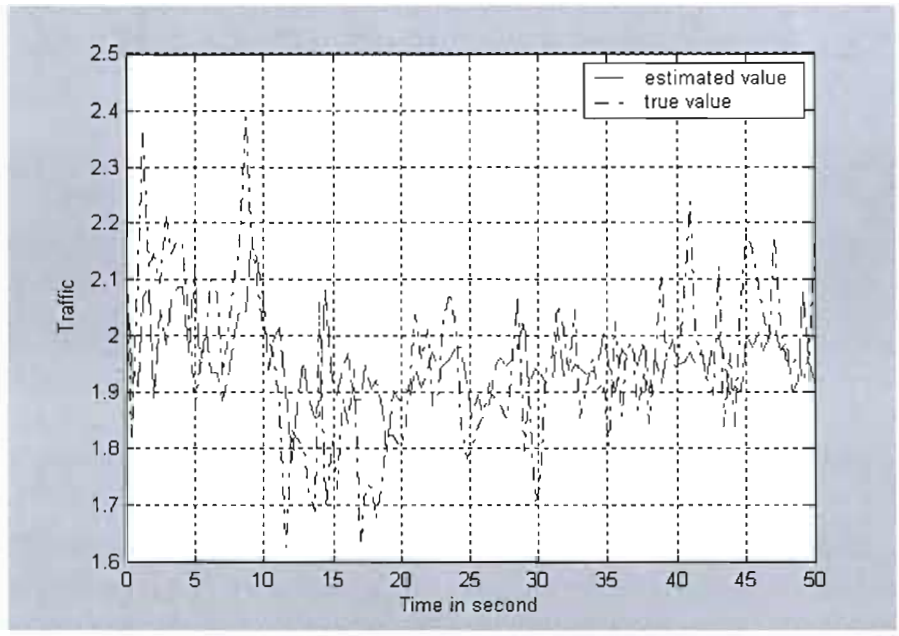
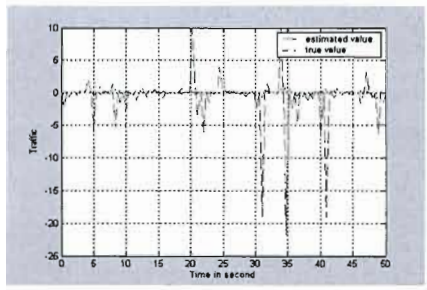


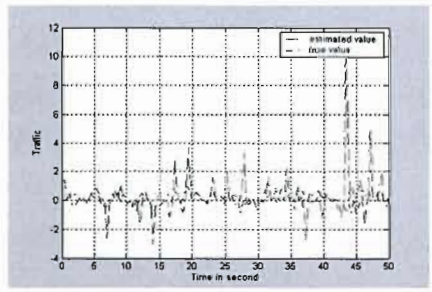
Figure 5.23 Trafic estimé VS Trafic réel avec LMMSE : in_trace, n=20

5.2.5.5 Influence du paramètre de Hurst *H* sur le taux d'erreur

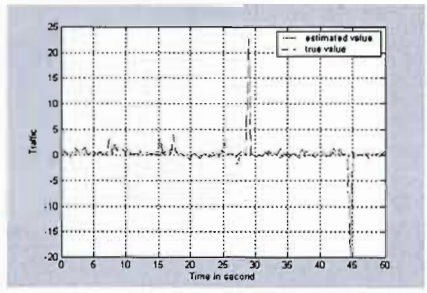
Afin d'étudier l'influence du paramètre de Hurst *H* sur le taux d'erreur commis par la prédiction avec le modèle *LMMSE*, on fait varier les valeurs de *H* (qui doivent être supérieures à *0.5*, pour conserver la propriété d'auto - similarité) de *0.60* à *0.90* en l'incrémentant à chaque fois de *0.05*. Pour chaque valeur de *H*, on simule un processus auto - similaire de paramètre *H* correspondant, et on obtient à chaque fois le graphe qui représente le trafic estimé par rapport au trafic réel, les figures suivantes (a), (b), (c), (d), (f), (g), (h) et (e) représentent les courbes pour la prédiction du trafic utilisant le modèle *LMMSE* avec des paramètres de Hurst *0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90* et *0.95* respectifs.



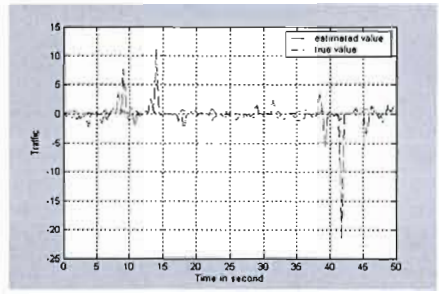
(a)



(b)



(c)



(d)

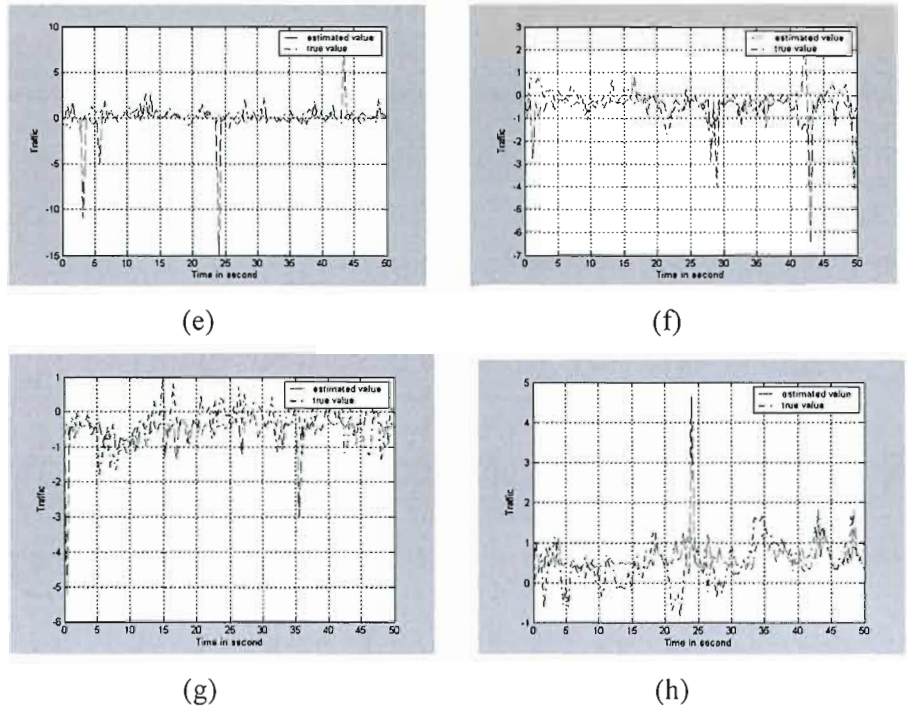


Figure 5.24 Trafic estimé VS Trafic réel pour H variés

Le tableau suivant montre bien la différence remarquable entre les taux d'erreurs obtenus à chaque fois qu'on fait varier la valeur de paramètre de Hurst *H*, d'après le tableau 5.19 ainsi que les courbes de la Figure 5.24, les meilleurs valeurs sont obtenues lorsque *H* varie entre *0.85* et *0.95*, les figures (f), (h) et surtout (g) illustrent bien nos observations, on pourrait dire que lorsque H augmente le taux d'erreur diminue et par conséquent on aura un trafic estimé qui ressemble bien au trafic réel, et par la suite on obtient une meilleure prédiction du trafic.

Tous les tests effectués, ont été fait avec des valeurs de $m = 14$ et $n = 20$. Le tableau contient les valeurs obtenues de l'écart type (la déviation standard) et pour l'erreur moyenne pour chaque paramètre de *H* correspondant :

Hurst	Ecart Type	Erreur Moyenne
0.60	3.4295	147.02%
0.65	1.2882	74.15%
0.70	2.7090	66.39%
0.75	1.2455	64.28%
0.80	1.8715	76.77%
0.85	0.4578	24.43%
0.90	0.5809	12.68%
0.95	0.8151	40.02%

Tableau 5.19 L'erreur moyenne de prédiction VS le paramètre H - LMMSE

La figure suivante représente la courbe représentative du tableau précédent et qui illustre la variation de taux d'erreurs par rapport au paramètre de Hurst H :

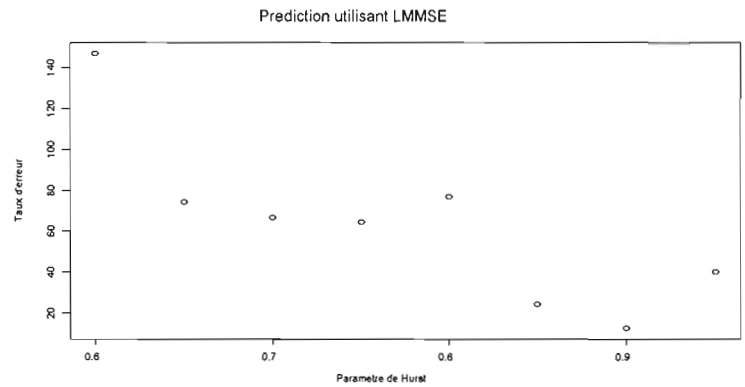


Figure 5.25 Influence de H sur le taux d'erreur avec LMMSE

5.2.5.6 Influence de la valeur de m et n sur le taux d'erreur

Faisons varier la valeur de *m* (la taille de la série de valeurs agrégées) et aussi de la valeur de *n* (l'horizon de prédiction) :

m	n	Erreur_Moyenne(%)	Ecart_Type
04	10	4.99	0.036
	20	4.81	0.037
	30	4.96	0.039
	40	4.99	0.038
14	10	3.8	0.031
	20	3.42	0.028
	30	3.43	0.029
	40	3.44	0.029
24	10	3.04	0.022
	20	2.96	0.024
	30	3.07	0.023
	40	3.11	0.023
34	10	2.88	0.023
	20	2.83	0.022
	30	3.17	0.022
	40	2.78	0.023
44	10	3.20	0.027
	20	3.33	0.028
	30	3.51	0.031
	40	3.53	0.028
	10	3.04	0.020

	20	3.56	0.025
	30	2.8	0.016
	40	***	***
	10	3.24	0.023
84	20	2.70	0.0222
	30	***	***
	40	***	***

Tableau 5.20 Influence de m et de n sur le taux d'erreur - LMMSE

D'après les valeurs données par le tableau, en variant respectivement les valeurs de *m* et de *n*, on remarque bien que les meilleurs taux d'erreurs sont obtenus lorsque $m = 34$ qui représente la taille d'une liste de valeurs. Dans notre cas, la liste originale est de taille égale *2000*, on aurait *59* intervalles et chaque intervalle contient *34* valeurs de la série chronologique initiale.

On remarque bien, que les bons résultats des taux d'erreurs sont donnés lorsque $n = 20$, qui représente, dans notre cas, l'horizon de prédiction ou encore le nombre de valeurs à prévoir dans le futur.

5.3 Comparaison des différents algorithmes utilisés

Pour des processus à longue mémoire, une bonne prédiction à long ou à court terme peut être obtenue à l'aide d'un grand nombre d'enregistrements pris du passé, autrement, il faudrait se baser sur un historique de taille assez grande.

Pour avoir une meilleure prévision, il faut se baser sur ces critères :

- bien choisir le modèle de prédiction pour fournir une grande exactitude

- afin d'achever une prévision en temps réel, il faut avoir une certaine simplicité dans le choix des paramètres à utiliser et à implémenter.
- la majorité des modèles de modélisation de trafic ont été fait avec des données off - line, on voudrait faire alors cette prédiction avec des données et des valeurs prises on - line.
- un bon modèle de prévision qui doit s'adapter à tout changement de trafic.

Pour concevoir le meilleur prédicteur qui permet de prédire un trafic Internet, plusieurs issus doivent être prises en considération :

- On doit déterminer la quantité de trafic initiale à mesurer où à partir de laquelle on pourrait appliquer des modèles pour le trafic future, dans notre cas, on se baserait sur des moyennes de quantité de trafic durant des intervalle de mesure (Eq-5.2.5.1), ce qui est due à la nature de dépendance à long terme (*LRD*) du trafic Internet qui implique une similarité de la fonction d'auto-corrélation entre la série des valeurs moyennes de trafic avec la série qui représente le trafic originale ; ce qui nous a permit de choisir la série agrégée pour représenter la série originale.

- Déterminer la bonne méthode de prédiction à utiliser, qui se base sur une fondation théorique solide basée sur des concepts mathématiques et statistiques et qui permet en même temps une implémentation facile avec un minimum de paramètres à estimer.

En pratique, il existe plusieurs modèles fractionnels qu'on pourrait utiliser et appliquer sur les séries chronologiques possédant la propriété de portée à long terme (*LRD*), parmi ces méthodes, on avait testé les modèles : *ARIMA*, *F-ARIMA* (*Fractal ARIMA*), *FBM* (*Fractional Brownian Motion*), et *LMMSE*. Et à partir des résultats obtenus, en comparant les modèles testés en prédiction de trafic, on a pu sortir avec des conclusions satisfaisantes. Cette comparaison été faite en terme de simplicité, d'exactitude et de la façon d'implémenter la méthode.

5.3.1 Façon d'implémentation

Pour implémenter le modèle prédicteur fractionnel *LMMSE*, il faudrait estimer on – line le paramètre de Hurst *H*, c'est-à-dire, estimer la valeur de paramètre de Hurst *H* à partir des valeurs que contient la série chronologique initiale. Ce qui représente un grand avantage par rapport aux autres modèles de prédiction (*FBM* et *F-ARIMA*) qui doivent par contre estimer la valeur du paramètre de Hurst *H* avant d'utiliser ces modèles en prédiction.

En ce qui concerne le prédicteur *FBM*, on doit calculer les coefficients du modèle, parmi ces coefficients, il y a un paramètre qu'on doit trouver de la manière suivante:

$$\frac{\sin(\pi(H-\frac{1}{2}))}{\pi}(-t(T+t))^{-H+\frac{1}{2}}\int_0^a \frac{(\tau(\tau+T))^{H-\frac{1}{2}}}{\tau-t}d\tau \qquad \text{(Eq-5.3.1).}$$

où T est l'intervalle dans lequel l'information de l'historique recueillie pour prévoir la valeur à l'instant $T + t$ (Eq-5.3.1), ce coefficient est très complexe, l'équation qui permet d'obtenir la valeur de son estimé est assez compliquée à implémenter.

Ou encore dans le modèle *F-ARIMA*, où on doit estimer la valeur de $d = H - ½$ puis calculer les coefficients β_{kj} par :

$$\beta_{kj} = -\binom{k}{j}\frac{\Gamma(j-d)\Gamma(k-d-j+1)}{\Gamma(d)\Gamma(k-d+1)}$$

où $\Gamma()$ est la fonction Gamma définie par : $\Gamma() = \int_0^\infty x^{a-1}e^{-x}dx$.

Par contre, dans le prédicteur *LMMSE*, on n'a pas ce genre de calcul à effectuer pour estimer les paramètres, autrement, il y a moins de calcul, il y aura juste à calculer directement les fonctions d'auto-covariance *R* à partir des mesures collectées

en utilisant l'équation (Eq-2.3.4-(4)). En plus, d'autres algorithmes peuvent être utilisés pour calculer les coefficients du modèle *LMMSE* à partir de l'équation (Eq-5.2.5.3).

5.3.2 Exactitude

Le critère d'exactitude représente le plus important critère dans le choix du prédicteur, ce choix doit être basé sur les valeurs obtenues des taux d'erreurs de prédiction. Soient $f(t), \overline{f(t)}, \hat{f}(t)$, qui représentent, respectivement, le trafic réel, le résultat de *f(t)* approprié suivant le modèle *F-ARIMA* ou *FBM*, et le trafic estimé par le modèle *LMMSE*.

L'erreur totale du prédicteur est donnée par [20]:

$$|\hat{f}(t+\tau) - f(t+\tau)| = |\hat{f}(t+\tau) - \bar{f}(t+\tau) + \bar{f}(t+\tau) - f(t+\tau)|$$

$$\leq |\hat{f}(t+\tau) - \bar{f}(t+\tau)| + |\bar{f}(t+\tau) - f(t+\tau)|$$

$$= err_{\,model} + err_{\,prédicteur}$$

En fait, l'erreur de la prédiction représente la somme de l'erreur du modèle approprié $err_{\,model}$, introduit par le trafic réel approprié dans le modèle, et de l'erreur de prédiction, $err_{\,prédicteur}$, introduit par le même prédicteur.

He, Gao et Hou, [20], ont montré que le deuxième terme pourrait être calculé analytiquement, par contre le premier terme est déterminé empiriquement.

Dans les trois modèles de prédiction *LMMSE*, *F-ARIMA* et *FBM*, si le paramètre de Hurst $H \dashrightarrow 0.85$, alors l'erreur de prédiction converge vers *0*, et elle représente la valeur minimale la plus possible, les trois courbes représentatives des trois modèles (le premier est modélisé par *F-ARIMA* et le deuxième est modélisé par *FGN* et le troisième représente un trafic réel auto - similaire) [19]-[24] se coupent en un seul point où $H=0.85$, d'où la figure suivante :

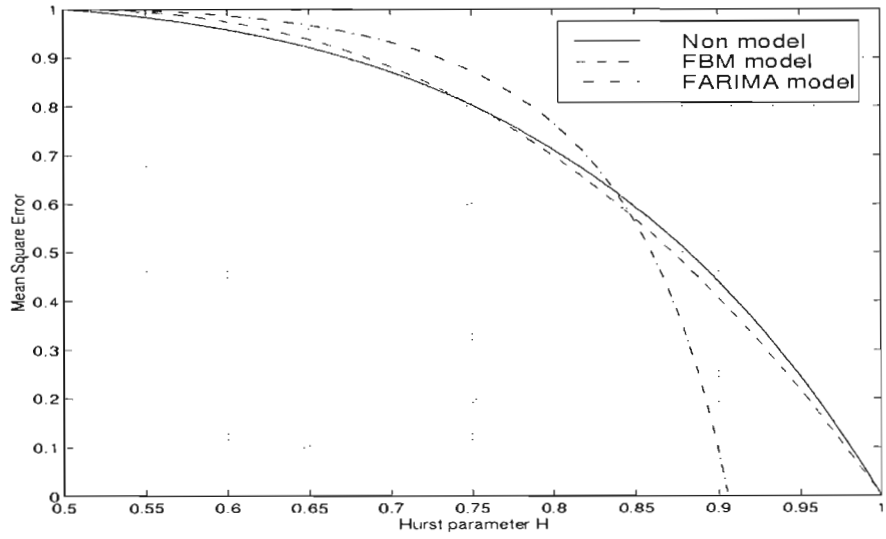


Figure 5.26 La valeur moyenne de l'erreur VS le paramètre de Hurst H

À partir de l'analyse faite sur les traces d'un trafic réel [15], le paramètre H du trafic Internet est souvent ≤ 0.85, et théoriquement, les trois prédicteurs sont relativement appropriés pour la prédiction du trafic Internet. Concernant les modèles *ARIMA* et *F-ARIMA*, on avait obtenu des taux d'erreurs variables, en générale assez acceptable de l'ordre de *25%* dans les meilleurs cas. Mais dans la plupart des cas, ce taux d'erreur est assez grand de façon qu'on ne puisse pas se baser sur un tel modèle qui ne fournit pas souvent les résultats désirés.

Le modèle *FBM* représente une grande complexité dans l'estimation de ces paramètres, on avait utilisé ce modèle pour la simulation et la génération d'un processus auto-similaire qui permet de donner une bonne représentation du trafic Internet réel.

En appliquant le modèle *LMMSE*, on a constaté la grande différence en prédiction, ce modèle fournit une grande exactitude, avec une grande ressemblance

entre le trafic réel et le trafic estimé par le modèle avec un taux d'erreur de l'ordre de *3%* dans les meilleurs cas. Aussi, ce modèle montre une grande simplicité dans son application ainsi que dans la façon d'implémenter le système.

CHAPITRE VI

CONCLUSION ET PERSPECTIVES

Les objectifs de mon travail de projet de fin d'études réalisé au sein du département téléinformatique peuvent être résumés ainsi :

* dans un premier temps, il était nécessaire de développer des outils de caractérisation et d'analyse du trafic réseau ;

* dans un deuxième temps, je devais mettre en place un outil performant permettant de prédire le trafic réseau à partir de traces réelles.

J'ai dû dans un premier temps me familiariser avec les anciennes et les nouvelles notions décrivant le trafic Internet. Ces derniers, pour les nouveaux modèles de caractérisation du trafic qui ont été mis en évidence dans les chapitres III et IV. Dans un deuxième temps, il a été nécessaire de préciser les notions nouvelles d'autosimilarité et de dépendance à long terme qui sont observables au travers des premières études et recherches observées sur le comportement de trafic Internet qui a été mis en évidence dans le chapitre IV.

Ensuite, j'ai pu introduire la contribution principale de mon travail en détaillant l'ensemble des outils mis en place : les outils analytiques et mathématiques qui sont utiles pour caractériser de façon précise le trafic, ainsi d'utiliser ces mêmes outils pour des fins de prédiction de trafic et de fournir l'algorithme idéale pour avoir une meilleure prédiction de trafic réseau à fin de bien connaître la bonne bande passante à affecter sur un lien pour ne pas avoir de congestion.

Dans le dernier chapitre de ce document, j'ai introduit aussi les principaux travaux déjà effectués dans ce domaine ainsi que les principales pistes de la recherche actuelle.

De plus, j'ai pu découvrir des notions nouvelles comme les concepts statistiques appliqués aux réseaux comme l'autosimilarité du trafic Internet ou les structures d'auto-corrélations observables dans le trafic Internet. Ensuite, j'ai dû appliquer une démarche de conception ingénieur nécessaire pour mettre en oeuvre l'ensemble des outils mathématiques et statistiques pour des fins de prédiction. Enfin, j'ai pu m'adonner à une activité de recherche autour du thème « Prédiction du trafic réseau » dont les grandes lignes ont été résumées dans le dernier chapitre de ce document.

Pour conclure, je dirais que ce projet m'a permis de profiter tant sur le plan humain que professionnel de l'ambiance d'un laboratoire de recherche. J'y ai découvert un milieu très intéressant où les gens sont passionnés par leur travail et très motivés. Mon impression sur ce milieu de la recherche est très positive.

BIBLIOGRAPHIE

[1] Konstantina Papagiannaki, Nina Taft, Zhi-Li Zhang, Diot "Long-Term Forecasting of Internet backbone Traffic: Observations and Initial Models", 2003

[2] Nancy K. Groschwitz and George C. Polyzos "A Time Series Model of Long Term NSFNET Backbone Traffic", 1995

[3] "A practical approach to forecast Quality of Service parameters considering outliers" Ilka Miloucheva, Eberhard Muller, Alessandro Anzaloni, 2003

[4] Halima Elbiaze, Omar Cherkaoui, Brenda Mcgabon, Michel Blais, "Exploiting Self-Similar Traffic Analysis in network resource control", 2003

[5] Leland, Taqqu, Willinger, Wilson, "On the Self Similar Nature of Ethernet Traffic (Extended Version)", IEEE/ACM Transactions on Networking, Fevrier 1994.

[6] Roberts, "Self-similar network traffic and performance evaluation", edited by K. Park and W Willinger, J. Wiley & Sons, 2000.

[7] D. Heyman, "Some issues in performance modeling of data teletraffic", Performance Evaluation, Vol. 34, pp. 227-247, 1998

[8] Beran, R. Sherman, M. S. Taqqu and W. Willinger, 'Long-range dependence in Variable-Bit-Rate video traffic', IEEE Trans. Comm., 1995.

[9] Leland, Taqqu, Willinger, Wilson, "On the Self Similar Nature of Ethernet Traffic", IEEE/ACM Transactions on Networking, Septembre 1993.

[10] M. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic : Evidence and possible causes", IEEE/ACM Trans. on Networking, Decembre 1997.

[11] Laurent Ferrara, Dominique Guégan, "Analyser les séries chronologiques avec S-PLUS, une approche paramétrique",France 2002

[12] Jan Beran, "Statistics for Long Memory Processes", Germany, 1994

[13] Yuan Gao, Guanghui He and Jennifer C. Hou, "On Exploiting Traffic Predictability in Active Queue Management", IEEE 2002

[14] A.M. Adas, "Using Adaptive Linear Prediction to Support Real-Time VBR Video Under RCBR Network Service Model"IEEE/ACM Transactions on Networking, Oct 1998.

[15] W. Willinger, V. Wilson, M. Taqqu: "Self-Similarity Through High-Variability: Statical Analysis of Etherent LAN traffic at the source Level" ACM, 1991

[16] Yantai Shu, Zhigang Jin, Lianfang Zhang, Lei Wang : "Traffic Prediction Using FARIMA Models"

[17] Nayera Sadek, Alireza Khotanzad, et Thomas Chen : "ATM Dynamic Bandwidth Allocation Using F-ARIMA Prediction Model"

[18] Timo Koski "Stochastiska ProcesserTAMS47/NMAC20 ", Feb 2003

[19] Guanghui He, Jennifer Hou, "On Exploiting Long Range Dependence of Network Traffic in Measuring Cross Traffic on an End-To-End Basis", Oct 2003

[20] Guanghui He, Jennifer Hou, Yuan Gao, Kihong Park: "A Case for Exploiting Self – Similarity of Network Traffic in TCP Congestion Control", 2002

[21] Majid Ghaderi, "On the Relevance of Self Similarity in Network Traffic Prediction", Août 2003.

[22] Z Duan, Z Zhang, Y Hou, "Service Overlay Networks: SLA's, QoS, and Bandwidth Provisionning", IEEE/ACM Transactions On Networking, Dec 2003.

[23]Petteri Mannersalo, "Some Notes On Prediction of teletraffic" Jan 2003

[24] lien Internet: http://lion.cs.uiuc.edu/courses/cs497hou/PAQM.pdf

[25] Pierre-Franc¸ois Quet Peng Yan Hitay Ozbay: "LMMSE Based Capacity Predictors for Flow Control in Communication Networks"

[26] Thèse : Abdel Haye, "Théorèmes limites pour des processus à longue mémoire saisonnière"

[27] Chadi Barakat¤, Patrick Thiran, Gianluca Iannaccone, Christophe Diot, Philippe Owezarski, "Modeling Internet backbone traffic at the flow level", Août 2003

[28] Larrieu, Owezarski "De l'utilisation des mesures de trafic pour l'ingénierie des réseaux de l'Internet", Juin 2002

[29] Chuck Fraleigh and Fouad Tobagi "Provisioning IP Backbone Networks to Support Latency Sensitive Traffic" , Juin 2003

[30] Padhye J., Firoiu V., Towsley D. and Kurose J., "Modeling TCP throughput : a simple model and its empirical validation",1998.

[31] Sikdar B. & Vastola K., "On the contribution of TCP to the selfsimilarity of network traffic", September 2001.

[32] http://www.cs.bu.edu/pub/barford/ss_lrd.html